LA-UR-02-7009
Approved for public release; distribution is unlimited.

Title:

Discrete Agent Modeling as a Tool For the Study of Individual and Social Development: The MICROS Code (Version 8.0)

Author(s):

Stephen M. Younger

Submitted to:

Discrete Agent Modeling as a Tool For the Study of Individual and Social Development: The MICROS Code (Version 8.0)

Stephen M. Younger
Los Alamos National Laboratory
Los Alamos, NM 87545

November, 2002

MICROS is a discrete agent simulation computer code designed to study the effect of alternate ethical systems on simple societies. It describes the time evolution of a population of agents following a rule-based decision process in a landscape containing food supplies, shelters, and material supply centers. Social organization is treated by assigning agents to home shelters with leaders who can be replaced by revolution or upon their death. Agents can be designated as good or evil. Good agents share information and goods while bad agents steal from others. An interaction matrix monitors the relative affection of agents toward one another. Reproduction and family units are simulated. Written in Visual Basic 6, the code is intended to be easily modified by the user.

I. Introduction

Simulating the effect of alternate ethical systems on individual and social development is of considerable interest to the philosopher and the social scientist. What is the effect of evil upon society? Do the good necessarily triumph over the bad? What are the effects of different leadership styles upon social productivity? Simple rule based models allow one to examine such questions in a controlled environment in which agent decisions are precisely defined.

Any simulation would be very hard pressed to represent the complexity of real human beings or even of higher animals. The similarity of the simulation to human society is an important, but not a dominant, component of this work. Rather, the value of such simulations rests in their ability to establish suggestive casual relationships between behavior norms and the resulting quality of life of an individual and the society in which that individual operates.

In the previous paper in this series, Younger (2002), a discrete agent simulation model called MICROS was described. MICROS calculated the time evolution of a population of agents in a landscape containing food supplies, shelters, and material supply centers. In this paper a modified version of that code, MICROS Version 8.0, is described in detail. Subsequent papers in this series will report the results of applying MICROS Version 8.0 to a variety of interesting social scenarios.

The MICROS landscape is divided into a square grid, similar to a chessboard. This grid contains three types of features: food centers, shelters and material centers. Agents with individual characteristics populate this world. Time is represented in steps. The agents are required to make decisions to satisfy hunger by visiting food centers, to rest at shelters, and to collect materials for the improvement of their home shelter, perhaps at the direction of a leader. An interaction matrix describes the relative affection of one agent toward another. Procreation is allowed and family units are simulated.

The agents satisfy five basic needs during each timestep of the simulation:

1. Sleep
2. Hunger
3. Companionship
4. Activity
5. Orders from their leader, if any.

Sleep requires resting some number of timesteps. This number varies according to whether the agent is at a shelter or in open terrain. Reduction of hunger requires consuming food resources at a food center, a shelter, or those carried by the agent itself or another agent in the same square. Companionship includes sharing knowledge or goods with another agent or mating with an agent of the opposite sex. Activity involves exploration of the landscape. If leaders are included in the simulation, they may issue orders to some or all of their population to bring food or materials to the home shelter which they command. If hunger or fatigue grow to unacceptable levels, then the agent will die. The agents have a maximum lifetime, so eventually they will die of old age. Typical simulations span many generations, so that equilibrium conditions can be established for the type of ethical system imposed on the artificial world.

II. Code Overview

General

MICROS is written in Visual Basic 6. It was designed to be easily modified by the user. Clarity of design was emphasized over efficiency.

An input file, microsin.txt (in txt format), contains most of the parameters of the simulation. This file must be located in the project file space. Output is written in text format to the files microsout.txt, microsdata.txt, and microssummary.txt. Microsout contains a summary of the input data, records of particular agent activities, and periodic snapshots of the overall status of the simulation. Microsdata contains key parameters of the simulation in a comma-delimited format suitable for reading by programs such as Microsoft Excel. Microssummary contains a summary of the input data and the details of the end state of the simulation along with an analysis of various time-averaged quantities. Graphs of population, quality of life, and the food and material reserves of the home shelters are presented in several windows on the startup form.

Pressing the Run button on the Visual Basic control panel or double clicking on the executable icon starts the code. This brings up the control panel of the simulation. Pressing the Start button begins the actual simulation.

Most of the simulation runs made so far using MICROS were relatively small, containing a few hundred agents at most. While the code can be expanded to handle much larger populations, the intent here was to allow for detailed study of individual agent decisions rather than large statistical ensembles.

Description of the Code Routines

Subroutine start_click

Subroutine start_click is the main program. It begins by setting various size-dependent variables, dimensioning and initializing the arrays, and reading the parameters of the simulation from the input file microsin.txt. A list of the variables in MICROS is given in Table I.

The initial direction of movement of each agent is set as a random number between 0 and 1 via the function rnd.

The initial leadership value of agent i is set as: act(i,18) =4* Int(Rnd * (maxpop + 1)).

The natural aggression factor of the agent is set as a random number between 0 and 1.

Agood is the fraction of good agents. An agent is bad if

Rnd > abs(agood).

Otherwise it is good. So if agood=0 all agents are bad and if agood =1 all agents are good. Negative values of agood mean that the initial distribution of good and bad agents specified in the input file is left unaffected – only newborn agents are assigned according to this formula. If agood>0 then the initial agents are also dealt with by this formula. The parameter shl(i,7) determines the morality of the agents in that home shelter. If shl(i,7) >0 then all agents in that shelter are good. If shl(i,7)<0 then all agents in that shelter are bad. If shl(i,7)=0 then a random distribution of good and bad is produced for newborn agents according to agood. The good/bad character of the initial agents is still determined by the agent cards.

The parameters of the simulation, including those set by the code itself, are written to the output files microsout and microssummary in text format. The date, time, and name of the simulation run are included in the heading of those files.

The simulation itself is executed by stepping through time in a for-next loop. One unit is added to the clock at each step. If clock exceeds the length of a day, dayl, then it is reset to 1.

The population is tallied. Only live agents are counted in the population.

The strength of the agent, act(i,4), is calculated as:

Strength = age – hunger - fatigue	for age <= lft/2
Strength = lft – age – hunger – fatigue	for age>lft/2

where lft is the agent lifetime.

The power factor, act(i, 21) is calculated as the sum of the natural leadership value of the agent plus the sum of the interaction matrix elements of the other agents toward the agent under consideration.

The social factor, act(i,24), is the sum of the interaction matrix elements of the other agents for the agent under consideration. It represents a degree of social acceptance of the agent.

Age, hunger, fatigue, need for companionship, and need for activity, are each increased by one unit.

The quality of life, act(i,9), is incremented by one unit. One point is deducted from the quality of life for each of the following conditions that apply:

If hunger>mxfd/2.
If sleep>mxsl/2.
If companionship>2*dayl
If activity>4*dayl

A set of statements evaluates the cause of death of any agents that died on the preceding timestep and prints this information in the output file. Array positions for agents who have died are reset for reuse by new agents.

The food centers are updated by the addition of their replenishment for that timestep. Food centers accumulate food at a fixed rate specified in the input file. There is no limit to the amount of food that can be accumulated.

The strength of each shelter is reduced by an amount dshl each timestep. Material collection can refurbish or build the strength of the shelter. These activities are controlled by the shelter leader assigning tasks to the shelter population.

If leader = true, then two leadership routines are called. First, the leader selection routine ldrselect is called. If no leader is designated in the input file or if the existing leader died then a leader is appointed. If a leader already exists, then the opportunity for leader replacement through revolution is evaluated. The second leader decision routine, ldrdecide, decides which priorities are assigned to the agents of the relevant home shelter – food collection or material collection.

At this point all of the updates to the landscape and the agents are complete and it is time to begin the individual agent decision processes. This is done in subroutine agentdec.

An analysis of various time-averaged quantities is performed. These are shown in Table 2.   Note that time averaged quantities are averaged over the agents and over time. Plots are prepared.

If the time corresponds to the interval designated for a printout of conditions, a listing of selected agent and landscape conditions is sent to the output file. The average values of the strength, the quality of life, the power factor, and the social factor of agents associated with the populations of good and bad agents are computed. In this version of the code up to two home shelters are permitted. The number of times that good agents shared and that bad agents stole as well as the number of times that they communicated and mated are tallied.

This completes the total simulation update for the timestep. The code goes to the next time, and checks to see if tstop, the time for ending the simulation has been reached. If so, the code does a final analysis of the simulation and plots the results.

Subroutine agentdec

Subroutine agentdec is the key routine of the simulation, the one in which the agents decide what to do during each timestep. The logic and sequence of these decisions are important to understanding the performance of the simulation. Usually, only one activity can occur per timestep. For example, if a female gives birth, that is the only activity permitted her on that timestep. However, if an agent eats, it can also collect food for its home journey. Agents can communicate, share, steal, and mate on the same timestep.

First, the code checks to see if the agent is female and at the end of her gestation period. If so, the communication subroutine is called so that birth can occur.

Second, if time has advanced far enough that births are possible and the age of the agent under consideration is less than lft/5 and the agent's hunger factor is less than 0.8 x mxfd then the position of that agent is constrained to be that of its mother. This represents a nurturing period for young agents. The young agent still decides its own actions – the only requirement is that it be collocated with its mother.

Third, if monog = true (monogamy exists) then the position of the female member of a mating couple is constrained to be collocated with her husband. Given the previous constraint on children, this results in the collocation of family units. Offspring older than lft/5 are free to go their own way. They are not required to remain with their parents.

If none of these constraints apply, then the priorities that govern agent decisions are:

1. Sleep
2. Food
3. Follow order from shelter leader
4. Companionship
5. Activity

The decision sequence used in MICROS are:

1. If sleep>3*mxsl/4 or ((sleep>hunger and sleep>companionship and sleep>activity) or (agent is sleeping)) and hunger<0.75*mxfd then call the sleep routine and end the timestep for that agent.
2. If (hunger>mxfd/2 or (hunger>companionship and hunger>activity)) then call the eating routine and end the timestep for that agent.
3. If the agent has an order from its leader or is carrying supplies home, then carry out that action and end the timestep for that agent.
4. Communicate with other agents present in that square of the landscape. If communication takes place, end the timestep for that agent.
5. If communication has not occurred and communication>activity then look for a nearby agent with whom to communicate and move toward that agent so that communication can take place on a future timestep. End the timestep for that agent.

6. At this point activity has emerged as the highest priority. Explore the landscape by movement.

Subroutine agenteat

Subroutine agenteat controls agent eating and searching for food.

If an agent is carrying food, then part or all of that food is consumed. If not, the routine checks to see if the agent is at its home shelter and if that the shelter has a food supply. If so, it consumes some or all of that food. If the agent is at a food center and the food center has food then the agent consumes some or all of that food and collects up to mxcy units (its maximum capacity for carrying) to take back to its home shelter. In any case, the maximum food that can be consumed is that required to reset the agent's need for food equal to zero. If food is collected, the first element of the agents knowledge array kno(i,1) is set to –1, indicating that it is carrying something and that its destination is its home shelter. The second and third elements of the agent's knowledge array are set to the x and y coordinates of its home shelter.

If the agent's hunger factor is greater than 0.65 x mxfd and there is food at the agent's home shelter, then the agent goes home.

If the agent is not carrying food and it is not at a place where there is food then it searches its memory for the location of the nearest food center that had food at the latest time of which it has knowledge. It then sets the second and third elements of its knowledge array to the coordinates of that center and calls subroutine agentmove to move one step toward it. If the agent does not know the location of any food center that currently has food then it will check to see if its home shelter has food and if so, it will go there. If the agent does not know the location of any food center with food and its home shelter does not have any food then the first element of its knowledge array is set to 8, indicating that exploration is to be performed that timestep, and the movement subroutine is called. (The movement routine contains sensation whereas agentmove only moves the agent towards a predetermined goal.)

Subroutine agentslp

Subroutine agentslp controls an agent sleeping or looking for a place to sleep.

If an agent is already sleeping then the time remaining during its sleep period is reduced by one and the subroutine is ended.

If the agent is not yet asleep then the time required for an agent to be refreshed (to have its need for sleep set to zero) depends on its location:

Home shelter:	6 timesteps
Non-home shelter:	8 timesteps

Open terrain: 10 timesteps

The routine checks to see if it is at a shelter and if so it puts the agent to sleep and sets array position act(i,11) as an "alarm clock" to indicate when it should wake up. When asleep the agent does not eat, communicate, or move. (Each of these needs is increased by one unit during sleeping.)

If the agent is not at a shelter, but its need for sleep is more than 90% of mxsl, the maximum need for sleep before death by exhaustion, then the agent is put to sleep on the spot.

If the agent is not at a shelter, but the need for sleep is less than 90% of mxsl then it searches its memory for the location of the home shelter and moves one step toward it; provided that the time required to reach the home shelter is less than one eighth of a day. If that distance is too far, then the agent finds the nearest non-home shelter and subroutine movement is called to move one step toward that shelter. If the agent does not know the location of any shelter then the first element of its knowledge array is set to 8, indicating that exploration is to be performed that timestep, and the movement subroutine is called. (The movement routine contains sensation whereas agentmove only moves the agent towards a predetermined goal.)

Subroutine agentcomm

Subroutine agentcomm governs agent communication and reproduction. Communication is treated first, and then reproduction.

Communication is attempted with any live agent, j, at the same location as agent i, that is awake, and that has an interaction matrix element imx(i,j) greater than or equal to 0. (A negative interaction matrix element indicates that agent i does not like agent j and hence does not wish to communicate with it.) If there is another agent present that meets these conditions the communications needs for both agents i and j are set to zero, even if no information is transferred. Agent i can communicate with any number of agents present at the same location on the same timestep. Only good agents give information. Bad agents can receive information from good agents.

Information is transferred by reviewing each fact in agent i's memory and checking to see if agent j knows that fact and, if the fact concerns a food or material center, whether the information is of a later time than what is included in its own memory. If agent j does not have this fact in its memory, it is put there. If agent j's information is older than agent i's, then agent j's information is updated. One point is added to the interaction matrix elements imx(i,j) and imx(j,i) , to the quality of life of agent i, and to the leadership value of agent i. A message indicating that communication or updating occurred is sent to the output file.

Facts are time dependent. When an agent senses a food or material center it stores the time and the quantity of goods at that center. This enables agents to share recent information about the supplies at food and material centers. Even though they may all know the locations of such centers, they are constantly updating one another on the quantities of good available at each.

If agent i is carrying something (food or materials), agent i is good, and there are other agents present, then it will share with those other agents. Again, the receiving agent must be alive, awake, and the interaction matrix element imx(i,j) must be greater than or equal to zero. Agent i shares equally with all agents in the square. The interaction matrix element imx(i,j) is increased by twenty percent of the amount shared and the interaction matrix element imx(j,i) is increased by ten percent of the amount shared. The quality of life of agent i is increased by twenty percent of the amount shared and the quality of life of agent j is increased by ten percent of the amount shared. (Thus following the adage that "It is better to give than to receive.") The leadership value of agent i is increased by one unit. A message indicating that sharing has occurred is sent to the output file. If sharing has occurred then kno(j,1) is set to –1, indicating that agent j is to return home to deliver what it is carrying. The need for companionship for both agents is set to zero.

If agent i is bad, and is not carrying anything, then it will attempt to steal from any agents, good or bad, in the same square who are carrying food or materials and whose strength is less than its strength. If these conditions are met then agent i will steal all of the belongings of agent j. Theft will only occur for the lowest j encountered, since at that time agent i will be carrying something and will not try to steal from other agents in the square. Ten percent of the quantity stolen is deducted from the quality of life of the agent stolen from. The interaction matrix element imx(i,j) is increased by ten percent of the amount stolen, indicating agent j as an "easy target." The interaction matrix element imx(j,i) is decreased by ten percent of the amount stolen, indicating agent j's fear of agent i. Once theft has occurred then kno(i,1) is set to –1, indicating that it is to return home to deliver what it is carrying to its home shelter. The need for companionship of agent i is set to zero.

Reproduction occurs for agents aged between one quarter and one half of their lifetime. This age requirement assures that reproduction does not occur between parents and offspring. Reproduction occurs only for agents of different sex in the same square with zero or positive interaction matrix elements for each other. If sameshl=true then only agents assigned to the same home shelter can mate with one another. If sameshl=false then agents of different home shelters can mate with one another. Birth occurs only after a gestation period of gest days. Once pregnant, the female cannot mate again until she has delivered her offspring. The probability of conception is 0.25 plus 0.05 if imx(i,j)>0 plus 0.05 if imx(j,i)>0 plus 0.05 if the agents are at their home shelter. If the monogamy control parameter monog is true, then once the female becomes pregnant, both male and female agents are considered permanently associated and cannot mate with other partners. If monog is false then male and female agents can mate freely with any agent of the opposite sex, so long as the female is not already pregnant. Upon first mating the quality of life and the interaction matrix elements of each agent are increased by 100 points. The quality of life and interaction matrix element of each agent is increased by 20 points for each mating, including the first. The need for companionship of each agent is set to zero.

The code checks to see whether there is an array position of a dead agent that can be filled. If not, nac is increased by 1. The sex of the new agent is set randomly as is its aggression factor and natural leadership factors. It is born at the location of the mother. Whether it is good or bad is determined according to the rules previously described. It is assigned the same home shelter as the mother. The knowledge of the mother is passed on to the offspring. Upon birth the interaction matrix element of each parent toward the child is increased by 100 points. The interaction matrix elements of the child toward each parent are increased by 90 points. All other interaction matrix elements for the new agent are set to zero. A message is sent to the output file.

Subroutine movement

Subroutine movement controls sensation and movement.

The routine first checks for the presence of agents, food centers, shelters, and material centers within sensory range. Sense range mxse applies to each direction independently so that an agent can sense a feature mxse away in each direction. When something is sensed its type and location are stored in the sns array. The agent then checks its memory to see if all of the newly sensed information is already in memory. If not, the new fact is stored in memory. If the fact is the location of a food or material center, the quantities stored in the center are recorded as well as the time. The locations of agents are not stored in memory, since they are constantly changing with time and since the affection or dislike of one agent toward another is stored in the interaction matrix. The leadership value of the agent is increased by one unit for each fact learned and the quality of life is increased by five points.

The agent then executes the movement instructions provided by other routines such as agenteat, agentslp, and ldrdecide. These instructions are stored in the first three elements of the agent's knowledge array:

Kno(i,1)= type of movement to be executed

-1	return to home shelter
-5	search for food
-6	search for materials
7	search for other agents with whom to communicate
8	explore

kno(i,2) = x-location of movement destination
kno(i,3) = y-location of movement destination

If kno(i,1)=-1 then the home shelter is the goal and subroutine agentmove is called and the agent moves one step homeward.

For the other negative values of kno(i,1) the agent checks to see if it is at a relevant center and, if so, it collects food or materials and sets its first knowledge array elements so as to return home. The activity need for the agent is decreased by one point.

If not at a center at this timestep, the agent searches its memory for the nearest food or material center that has nonzero contents and moves one step toward that center. If no such center is known, then a random direction of movement is chosen and the agent is moved one step.

If companionship is the goal, then the agent searches its sensation array, sns, for the location of the nearest agent and moves one step toward that agent. If no agents are sensed, then a random direction of movement is chosen and the agent moves one step in that direction.

If activity is the goal, then the agent moves one step in the direction stored in act(i,14). The need for activity is decreased by one unit. (A second unit is deducted in subroutine agentmove for a total of two units per timestep spent exploring.) The quality of life is increased by one unit for each timestep spent exploring.

In all cases of movement the code checks to see if the boundary of the landscape has been reached. If so, another random direction of movement is chosen and the agent is moved one step.

Subroutine agentmove

Subroutine agentmove moves the agent one step toward the goal stored in the second and third location of the agent's knowledge array. One point is deducted from the need for activity. If the agent is at its home shelter and is carrying anything then it deposits its cargo and its activity need is set to zero, indicating that it has accomplished something. Materials deposited at a home shelter increase the strength of that shelter.

Subroutine ldrselect

Subroutine ldrselect governs the selection of a leader and its replacement by revolution.

If leader=true, and there is no leader for a shelter with a population of three or greater, then the agent with the highest leadership factor, act(i,18), who is associated with the shelter under consideration, and who is more than lft/5 timesteps old, is chosen as leader. Upon appointment, the interaction matrix element of the leader toward all other agents is doubled to reflect caring of the leader for agents with whom it has had positive experiences and distrust of those with whom it has had negative experiences. The interaction matrix elements of all agents (associated with that shelter) for the leader are doubled for similar reasons. The power factor of the leader is increased by the population of the shelter + (food at shelter/(10 x mxfd x population) + (strength of shelter/(1000 x population)).

At any timestep the appointed leader can be replaced by a revolution. If another agent associated with the shelter under consideration has a greater power factor than the existing leader then that agent replaces the existing leader. The power factor is defined as the sum of the leadership factor of the agent and the sum of all of the other agents' interaction matrix elements toward that agent.

Subroutine ldrdecide

Subroutine ldrdecide governs the orders given by a shelter leader to the population of that shelter. A leader remains at its home shelter unless looking for food for itself.

If there is a leader for the shelter and clock is within the workday and clock is less than the leadership aggression factor times dayl and the leader knows something then an order is given. The rationale for keeping the leadership aggression factor times the length of day within bounds is to allow for weak leaders or exceptionally strong leaders. Note that there is still the proviso that the time be within the workday. Workday can be changed to permit even more aggressive cultural norms for work.

The only orders permitted in this version of MICROS are food and material collection. The base food reserve required at a shelter is 1.3 times the shelter population times the maximum food need, mxfd. If this requirement has not been met then the leader orders the agents awake at the home shelter to collect food. The leader selects the nearest food center within its memory and sends the agents to that location. Note that since agents will communicate with the leader upon their return from exploration or collection, the leader may well have more knowledge that it would have if left to its own devices.

If the food requirement has been met then the leader will order the agents who are at the home shelter and who are awake to collect materials. The leader selects the nearest material center and sends the agents to that location.

To reflect goodwill toward a leader who has met the minimum food requirement at the home shelter the interaction matrix elements of the agents assigned to that shelter relevant to the leader are increased by the ratio of the supply of food to the food requirement. The same is done for the material requirement.

Plot subroutines

Four subroutines plotpop, plotqol, and plotshlm, and plotshlf plot the population, the average quality of life of agents associated with each home shelter, the material resources of each home shelter, and the food supply available at each shelter. Black refers to total population. Red and blue correspond to the two home shelters. Yellow corresponds to bad agents and green to good agents.

Subroutine plotland plots the locations of agents, shelters, food centers, and material centers at the end of the run. Numbers correspond to the number of agents at that location. Red dots denote shelters. Green dots denote food centers. Blue dots denote material centers.

III. Variables

The variables used in MICROS are given in Table 1. Most variables are set in the input file. Some, such as array sizes and parameters involved in the decision processes of the agents, are set in the code itself. The user can adjust the variables to achieve considerable flexibility in the simulations.

Time, t, is measured in arbitrary units and is incremented by one unit per timestep. One unit of time is roughly equal to one hour of real time.

Physical space is divided into squares which are the of a size that an agent could cross in one timestep, about two miles in human terms. A typical small simulation has sizewld set to 20. Too small a world results in unrealistic crowding. Too large a world results in infrequent agent interactions. For many small studies an initial population of 20 in a 20 x 20 world was found effective. Agents are not permitted to leave the landscape – it represents an island geography.

The maximum population allowed, maxpop, must be chosen as a compromise between size, runtime, and the desire to allow for population growth in the scenario. A value of maxpop = 100 was found convenient for many applications. Note that the code does not stop when maxpop is reached as a result of procreation. The birth is simply ignored and an error message is sent to the output file. Caution should be exercised when maxpop is exceeded as the simulation is trying to execute a population explosion that is denied by the code. The maximum population that is allowed in the run due to the dimensioning of the array act is maxact. Thus maxpop must be less than or equal to maxact.

The number of agents, nac, is initially set in the input file but changes as births and deaths occur. The initial value must be sufficient to start a sustainable population but not so large that later population growth makes the simulation too large. Too small an initial number of agents does not enable sufficient procreation to occur. Too large an initial value of nac combined with too small a value of sizewld results in early runaway population growth. A value of nac = 20 was found convenient for small simulations.

The number of food centers, nfd, must be set to balance the need for agents to explore their environment and the satisfaction of their needs over time. Setting nfd = 5 with sizewld = 20 was found effective. The number of food centers should be adjusted based on their replenishment rate and the maximum population that is to be sustained.

Material centers are an aggregate representation of natural resources in the simulation. Materials are not required to sustain life and the number and characteristics of material centers can be adjusted to suit the aim of the simulation. Usually material centers are supplied with very large initial stocks and no increment is made over time. In scenarios designed to study aggression and acquisition these numbers can be reduced to force theft from other shelters.

The number of shelters is arbitrary. Shelters enhance the rate at which a rested state is achieved but they are not essential to survival. Home shelters are identified by parameters in the act array. Home shelters can have a leader if the parameter leader is set to true.

The lifetime of the agents is lft. It is often desirable to set lft to values as small as 1,000 to 10,000 timesteps to shorten generation time. Setting lft too small may not allow an agent to adequately explore its environment to find required food supplies. Setting lft to 4000 was found useful for many small simulations.

The length of a day is dayl and is usually set to 24. The length of a day governs sleep requirements and how much work can be performed for a leader upon command.

The maximum food need permitted before an agent starves is mxfd. A value of 200 is convenient for many simulations. This corresponds to just over eight days of human time. Setting mxfd too small results in starvation of agents, especially if they are in a large landscape with few food centers. Setting it too high reduces the survival stress on the agent.

The maximum sleep need is mxsl. A value of 100 is convenient for many simulations. This corresponds to just over four days of human time. Setting mxsl too low results in too much time spent sleeping compared to other possible actions. Setting it too high has few negative consequences except for unrealistically long periods of activity.

The range of sensation is mxse. A value of 2 is convenient for many simulation purposes. If one square on the landscape corresponds to two miles then sensation range is four miles in each direction. Note that mxse applies in each direction so that an agent at $x = 1$, $y = 1$ can sense another agent at $x = 3$, $y = 3$.

The maximum quantity of goods, food or material that an agent can carry is mxcy. Since one unit of food sustains an agent for one timestep, a setting of mxcy = 400 means that the agent can carry over sixteen days of food. Material resources are more arbitrary, but again a setting of 400 has been found reasonable for many small simulations.

The maximum number of facts that can be stored in an agent's memory array is knomax. Since the agents can only remember the locations of food centers, shelters, and material centers, knomax can be set to knomax = nfd + nsh + nmt. Since the array kno must be dimensioned in the code itself, knomax must be explicitly specified rather than automatically set by parameters in the input file.

The maximum number of sensations possible is snsmax. Since the agents can only sense other agents, food centers, shelters, and material centers, snsmax can be set to snsmax = maxpop + nfd + nsh + nmt. Since the array sns must be dimensioned in the code itself snsmax must be explicitly specified rather than automatically set by parameters in the input file.

The strength of each shelter deteriorates at the rate of dshl units per time. This is an arbitrary rate that can be adjusted to force more attention to be paid to shelter conditions by leaders. A rate of dshl = 1 was found reasonable for many small simulations.

The length of the workday is workday. This controls the length of time that an agent will spend carrying out the instructions of its leader. A very long workday can result in excessive deaths due to exhaustion or hunger. A value of workday = 12 was found convenient for many small simulations.

The length of the simulation in timesteps is tstop. This parameter should be chosen as large as possible to ensure that a stable situation has been reached in the simulation. It is normal to set tstop to at least to ten times the agent lifetime.

Setting leader = true enables leadership functions to occur in the simulation. Setting leader = false means that no leadership functions will occur.

Results of the simulation including some agent parameters and the conditions of food centers, shelters, and material centers, are printed every dtdump timesteps in the simulation. It is often convenient to set dtdump to 10% of an agent lifetime.

The parameter gest is the gestation period, the number of timesteps between mating and the birth of a new agent.

The switch monog determines monogamy in the simulation. If monog=true then a pregnancy causes agents to be coupled for the life of either agent. If monog=false then the agents can mate freely with anyone of the opposite sex provided age and other constraints, mentioned above, are satisfied. If one partner in a monogamous relationship dies, the surviving partner is free to mate with another agent.

A list of the time averages computing during the course of the simulation is given in Table 2.

The methods used for the calculation of all agent variables stored in act(i,j) are shown in Table 3.

Table 1. MICROS Variables

General variables

t	Time
sizewld	Dimension of the physical space
maxact	Maximum population allowed due to dimension of act
maxpop	Maximum population allowed in this simulation run
nac	Number of agents
nfd	Number of food centers
nmt	Number of material centers
nsh	Number of shelters
lft	Lifetime of an agent in timesteps
dayl	Length of a day in timesteps
mxfd	Maximum hunger before agent death
mxsl	Maximum fatigue before agent death
mxse	Range of sensation
mxcy	Maximum capacity for an agent to carry food or materials
knomax	Maximum number of facts permitted in the simulation
snsmax	Maximum number of sensations
dshl	Deterioration of a shelter per timestep
workday	Length of workday
tstop	Stop time of simulation
agood	Fraction of the population that is bad
Leader	If leader = true then allow a leader If leader = false then do not allow a leader
Dtdump	Print landscape and agent variables every dtdump timesteps
Gest	Gestation time for pregnancy
Monog	If monog = true then agents mate for life If monog = false then agents mate freely
Sameshl	If sameshl = true then only agents of same shelter can mate If sameshl = false then agents of different home shelters can mate

Arrays

act(i,j) — Description array for agent i

i	Agent identifier index
j	Parameter
1	Agent age in timesteps
2	X-location
3	Y-location
4	Strength
5	Hunger
6	Fatigue
7	Need for companionship
8	Need for activity
9	Quality of Life
10	Set to 0 if alive, set to 9 if dead
11	Sleep steps remaining
12	Home shelter number
13	Sex (0=female, 1=male)
14	Direction of movement
15	Object carried (2=food, 4=materials)
16	Quantity being carried
17	Good/evil switch (0=evil, 1=good)
18	Leadership value
19	Natural aggression factor
20	Gestation time left
21	Power factor
22	Mate identifier
23	Mother identifier
24	Social factor

sns(i,j) — Sensation array for agent i

j	Parameter
1	Type of object sensed 1 = agent 2 = food center 3 = shelter 4 = material center
2	x-location
3	y-location
4	amount of food or material in center

kno(i,j) Agent memory array for agent i

j	Parameter
knomax	Number of facts stored
1	Current task
	1 = food
	2 = sleep
	7 = companionship
	8 = activity
	-1 = return home
	-5 = collect food
	-6 = collect materials
2	x-location of current goal
3	y-location of current goal
5 to knomax-1	Parameters of facts
	5*n = type of object
	2 = food center
	3 = shelter
	4 = material center
	5*n+1 = x-location
	5*n+2 = y-location
	5*n+3 = quantity
	5*n+4 = time sensed

imx(i,j) Interaction matrix element of agent i toward j

fdc(i,j) Characteristics of food center i

j	Parameter
1	x-location
2	y=location
3	Quantity of food
4	Replenishment rate

mtc(i,j)	Location and description of material center i		
		J	Parameter
		1	x-location
		2	y-location
		3	quantity of materials

shl(i,j)	Location and description of shelter i		
		J	Parameter
		1	x-location
		2	y-location
		3	Strength
		4	Food present
		5	Leader
		6	Population
		7	Agent morality 0=random as per agood 1=good -1=bad

tplot(i)	Times of points plotted (Related to dtdump)
popshl(i,j)	Population of shelter I at time j
popplot(j)	Total population at time j
shlhm(i)	Shelter number of two home shelters
qolavg(i,j)	Average quality of life of agents in shelter I at time j
social(i)	Average quality of life of agents in shelter I
shlf(i,j)	Food supply of shelter I at time j
shlm(i,j)	Material supply of shelter I at time j
numgood(j)	Number of good agents at time j
numbad(j)	Number of bad agents at time j
qolgdbd(i,j)	Quality of life of good (i=1) or bad (i=2) agent at time j

actgdbd(k)	k=1: Number of good agents that communicated this step k=2: Number of bad agents that communicated this step k=3: Number of good agents that shared this step k=4: Number of bad agents that stole this step k=5: Number of good-good agent matings this step k=6: Number of good-bad agent matings this step k=7: Number of bad-bad agent matings this step

Table 2. Quantities Evaluated by MICROS and Printed in microsout and microssummary

Avgnumgood	Time average of number of good agents
Avgnumbad	Time average of number of bad agents
Avgqolgd	Time average of quality of life of good agents
Avgqolbd	Time average of quality of life of bad agents
Avgstrgd	Time average of strength of good agents
Avgstrbd	Time average of strength of bad agents
Avgpwrgd	Time average of power factor of good agents
Avgpwrbd	Time average of power factor of bad agents
Avgsocgd	Time average of social factor of good agents
Avgsocbd	Time average of social factor of bad agents
Avgcommgd	Time average of number of communications of good agents
Avgcommbd	Time average of number of communications of bad agents
Avgshrgd	Time average of number of times a good agent shared
Avgstlbd	Time average of number of times a bad agent stole
Avgmategdgd	Time average of number of matings involving two good agents
Avgmategdbd	Time average of number of matings involving one good and one bad agent
Avgmatebdbd	Time average of number of matings involving two bad agents

Table 3. Methods Used to Calculate Agent Parameters in act(i,j)

act(i,1)	Age	Increment one unit per timestep
act(i,2)	x-location	
act(i,3)	y-location	
act(i,4)	Strength	act(i,4) = age – hunger - fatigue for age <= lft/2 act(i,4) = lft – age – hunger – fatigue for age>lft/2
act(i,5)	Hunger	Increased one unit per timestep Decreased by amount of food consumed Minimum value is zero
act(i,6)	Sleep	Increased by one unit per timestep if awake Set to zero once agent is asleep
act(i,7)	Companionship	Increased by one unit per timestep Set to zero if another agent is present who is awake
act(i,8)	Activity	Increased by one unit per timestep Decreased by one unit if goods collected at food or material center Decreased by one unit if exploring Set to zero if goods deposited at home shelter Decreased by one unit whenever the agent moves
act(i,9)	Quality of Life	Increased by one unit per timestep Decreased one unit if hunger>mxfd/2 Decreased one unit if sleep>mxsl/2 Decreased one unit if companionship > 2*dayl Decreased one unit if activity > 4*dayl Increased five units if new fact sensed Increased one unit if fact given or received Increased by twenty percent of goods shared Increased by ten percent of amount of goods received in sharing event Decreased by ten percent of goods stolen Increased by 100 units on first mating if monog=true Increased by twenty units per mating Increased by ten percent of goods deposited at home shelter Add one unit if exploring

act(i,10)	Alive/Dead	Set to 9 if an agent is dead
act(i,11)	Sleep Steps	Decrease by one unit per step agent sleeps
act(i,12)	Home Shelter Number	
act(i,13)	Sex	Female = 0 Male = 1
act(i,14)	Movement Direction	Random number 0 –1.
act(i,15)	Object Carried	Food = 2 Materials = 4
act(i,16)	Quantity Carried	
act(i,17)	Good/Evil Switch	Good if rnd<abs(agood) Bad if rnd>abs(agood) Initial agents changed if agood>0 Initial agents as specified in microsin if agood<0
act(i,18)	Leadership Value	4*Int (rnd * (maxpop +1)) at start of run Increased by one unit when any amount shared Increased by one unit if fact learned by sensation Increased by one unit if fact given to another
act(i,19)	Aggression Factor	Random number 0-1 chosen at agent birth
act(i,20)	Gestation Remaining	Reduced by one unit per timestep
act(i,21)	Power Factor	Sum of leadership value and sum of imx(j,i) over j Upon selection as leader, increased by the population of the shelter + (food at shelter/(10 x mxfd x population)) + (strength of shelter/(1000 x population)).
act(i,22)	Mate Identifier	Agent number of agent i's mate
act(i,23)	Mother Identifier	Agent number of agent i's mother
act(i,24)	Social Factor	Sum of imx(j,i) over j

imx(i,j)	Increased one point per fact given to another Increased one point per fact received from another Increased by ten percent of goods received Increased by twenty percent of goods given to another Decreased by ten percent of good taken from another Decreased by ten percent of goods taken from agent i Increased by one hundred points for first mating Increased by twenty points per mating Set to 100 for offspring Set to 90 for parents

IV. Description of the Input File

The simulation reads the text file microsin.txt for input. All entries on the same line are column delimited. A sample input file is shown in Table 4.

The input file should contain:

name of run
sizewld
maxpop
nac
nfd
nmt
nsh
lft
dayl
mxfd
mxsl
mxse
mxcy
dshl
workday
tstop
agood
leader
dtdump
gest
monog
sameshl

For each food center:

fdc(i,1),fdc(i,2),fdc(i,3),fdc(i,4)

For each material center:

mtc(i,1),mtc(i,2),mtc(i,3)

For each agent:

act(i,1),act(i,2), act(i,3), act(i,4), act(i,5), act(i,6), act(i,7), act(i,8), act(i,9), act(i,10), act(i,11), act(i,12), act(i,13), act(i,17)

For each shelter

shl(i,1),shl(i,2),shl(i,3),shl(i,4),shl(i,5),shl(i,7)

Table 4. Sample Input File for Simulation with 20 Agents

```
Prototype Civilization Scenario
21
200
20
5
6
10
4000
24
200
100
2
400
1
12
40000
.5
true
100
200
true
false
3,3,100,20
6,16,100,20
11,11,100,20
16,6,100,20
19,19,100,20
11,3,1000000
3,11,1000000
19,11,1000000
11,19,1000000
3,19,1000000
19,3,1000000
0,2,8,100,0,0,0,1,0,0,0,3,1,1
0,7,15,100,0,0,0,1,0,0,0,3,1,1
0,9,12,100,0,0,0,1,0,0,0,8,1,1
0,5,3,100,0,0,0,1,0,0,0,8,0,1
0,18,17,100,0,0,0,1,0,0,0,3,1,1
0,4,12,100,0,0,0,1,0,0,0,8,1,1
0,19,9,100,0,0,0,1,0,0,0,8,0,1
0,12,7,100,0,0,0,1,0,0,0,3,0,1
0,18,20,100,0,0,0,1,0,0,0,3,0,1
0,10,11,100,0,0,0,1,0,0,0,8,1,1
0,2,8,100,0,0,0,1,0,0,0,3,0,1
0,7,15,100,0,0,0,1,0,0,0,3,1,1
0,9,12,100,0,0,0,1,0,0,0,8,0,1
0,5,3,100,0,0,0,1,0,0,0,8,1,1
0,18,17,100,0,0,0,1,0,0,0,3,0,1
0,4,12,100,0,0,0,1,0,0,0,8,1,1
0,19,9,100,0,0,0,1,0,0,0,8,0,1
0,12,7,100,0,0,0,1,0,0,0,3,1,1
0,18,20,100,0,0,0,1,0,0,0,3,0,1
0,10,11,100,0,0,0,1,0,0,0,8,0,1
```

3,9,0,0,0,0
16,11,0,0,0,0
6,6,100,0,2,0
9,3,0,0,0,0
19,13,0,0,0,0
13,19,0,0,0,0
11,6,0,0,0,0
16,16,100,0,3,0
6,11,0,0,0,0
11,16,0,0,0,0

V. Sample Run

Part of the output file, microsout, describing the initial part of a run using the input file contained in Table 4 is given in Table 5. The summary file microssummary is given in Table 6. The input parameters are repeated at the start of the file. At each dtdump timestep a summary of the landscape and agent conditions is provided. Note that mating does not occur until lft/4. As time progresses in this scenario (with the leadership switch set to true) the agents collect food and materials, indicated by the rising values of these quantities at the home shelters. Revolutions occur from time to time as some agents become stronger than the current home shelter leader. Deaths occur due to hunger and old age. In this run the food centers were purposely kept at a low resupply rate so that the agents could die of hunger or even of exhaustion if they were driven to find food at the expense of rest. Communication is frequent, including sharing of goods and information among agents with positive interaction matrix elements and stealing by agents whose good/bad switch is set to "bad."

The screen containing the graphics is shown in Figure 1.

Table 5. Initial Output from Run with Input File Shown in Table 4.

Micros Social Simulation

Run Prototype Civilization Scenario with NEW 21x21 landscape

Date 09-07-2002 Time 16:00:18

Size of Landscape= 21
Maximum population= 200
Number of agents= 20
Number of Food Centers= 5
Number of Material Centers= 6
Number of shelters= 10
Actor Lifetime= 4000
Length of Day= 24
Maximum Food Need= 200
Maximum Sleep Need= 100
Maximum Sensory Range = 2
Maximum Carry Capacity= 400
Shelter Deterioration Rate= 1
Workday= 12
Stoptime= 40000
Goodness parameter= 0.5
Leadership allowed:true
dtdump= 100
Gestation Period= 200
Monogamy: true
Mate with only those of same shelter false

Shelters

3	9	0	0	0		0
16	11	0	0	0		0
6	6	100	0	2	10	0
9	3	0	0	0		0
19	13	0	0	0		0
13	19	0	0	0		0
11	6	0	0	0		0
16	16	100	0	3	10	0
6	11	0	0	0		0
11	16	0	0	0		0

Food Centers

3	3	100	20
6	16	100	20
11	11	100	20
16	6	100	20
19	19	100	20

Material Centers

11	3	1000000
3	11	1000000
19	11	1000000
11	19	1000000
3	19	1000000
19	3	1000000

Agents
0 2 8 100 0 0 0 1 0 0 0 3 1 0.2741053 1 224 0.9777883
0 7 15 100 0 0 0 1 0 0 0 3 1 0.4176249 0 104 0.8363596
0 9 12 100 0 0 0 1 0 0 0 8 1 0.2468173 1 292 0.2175482
0 5 3 100 0 0 0 1 0 0 0 8 0 0.9436283 1 600 0.8838664
0 18 17 100 0 0 0 1 0 0 0 3 1 0.4956676 0 508 0.8488029
0 4 12 100 0 0 0 1 0 0 0 8 1 0.1493344 0 476 0.454948
0 19 9 100 0 0 0 1 0 0 0 8 0 0.1129414 0 256 0.7021187
0 12 7 100 0 0 0 1 0 0 0 3 0 0.5098409 0 596 0.8254834
0 18 20 100 0 0 0 1 0 0 0 3 0 0.5815489 0 368 0.1236871
0 10 11 100 0 0 0 1 0 0 0 8 1 0.1408705 1 276 0.0369764
0 2 8 100 0 0 0 1 0 0 0 3 0 0.2750251 1 356 0.4753245
0 7 15 100 0 0 0 1 0 0 0 3 1 0.2987708 0 104 0.396556
0 9 12 100 0 0 0 1 0 0 0 8 0 0.9575302 1 384 0.6344722
0 5 3 100 0 0 0 1 0 0 0 8 1 0.8805146 0 732 0.976976
0 18 17 100 0 0 0 1 0 0 0 3 0 0.2838494 1 76 0.4941967
0 4 12 100 0 0 0 1 0 0 0 8 1 0.9236993 0 60 0.1166154
0 19 9 100 0 0 0 1 0 0 0 8 0 0.2993929 0 532 0.4631894
0 12 7 100 0 0 0 1 0 0 0 3 1 0.1065478 0 64 0.919478
0 18 20 100 0 0 0 1 0 0 0 3 0 0.9401957 1 696 0.9657667
0 10 11 100 0 0 0 1 0 0 0 8 0 0.2479076 0 788 0.7551928

Revolution at shelter 3 with agent 19 replacing agent 2 at time= 1
Revolution at shelter 8 with agent 20 replacing agent 3 at time= 1
Food Center 3 was emptied at t= 6
Food Center 5 was emptied at t= 7
Agent 11 communicated information to agent 17 at t= 20
Agent 19 shared 116 with agent 9 at t= 23
Agent 10 communicated information to agent 6 at t= 30
Agent 10 communicated information to agent 16 at t= 30
Agent 10 shared 71 with agent 6 at t= 30
Agent 10 shared 71 with agent 16 at t= 30

t= 100 Population= 20 Births= 0 Deaths = 0

Number of good agents= 8
Average quality of life of good agents this timestep= 44.45
Average strength of good agents this timestep=-9.25
Average power factor of good agents this timestep= 370.35
Average social factor of good agents this timestep= 4.1

Number of bad agents= 12
Average quality of life of bad agents this timestep= 33.4
Average strength of bad agents this timestep=-0.916666666666667

Average power factor of bad agents this timestep= 388.716666666667
Average social factor of bad agents this timestep= 4.88333333333333

Shelters

3	9	0	0	0	
16	11	0	0	0	
6	6	0	0	19	10
9	3	0	0	0	
19	13	0	0	0	
13	19	0	0	0	
11	6	0	0	0	
16	16	0	0	20	10
6	11	0	0	0	
11	16	0	0	0	

Food Centers

3	3	2100	20
6	16	2100	20
11	11	1880	20
16	6	1673	20
19	19	1860	20

Material Centers

11	3	1000000
3	11	1000000
19	11	1000000
11	19	1000000
3	19	1000000
19	3	1000000

Agents
i= 1 x= 16 y= 6 Hunger= 73 Fatigue= 0 Companionship= 98 Activity= 73 QOL= 41
kno(i,1)= 7 kno(i,2)= 3 kno(i,3)= 9 kno(i,knomax)= 2
i= 2 x= 11 y= 15 Hunger= 100 Fatigue= 0 Companionship= 100 Activity= 21 QOL= 17
kno(i,1)= 7 kno(i,2)= 11 kno(i,3)= 16 kno(i,knomax)= 2
i= 3 x= 9 y= 12 Hunger= 100 Fatigue= 0 Companionship= 100 Activity= 25 QOL= 9
kno(i,1)= 7 kno(i,2)= kno(i,3)= kno(i,knomax)=
i= 4 x= 5 y= 4 Hunger= 100 Fatigue= 0 Companionship= 100 Activity= 25 QOL= 9
kno(i,1)= 7 kno(i,2)= kno(i,3)= kno(i,knomax)=
i= 5 x= 19 y= 17 Hunger= 100 Fatigue= 0 Companionship= 100 Activity= 25 QOL= 9
kno(i,1)= 7 kno(i,2)= kno(i,3)= kno(i,knomax)=
i= 6 x= 4 y= 12 Hunger= 69 Fatigue= 0 Companionship= 70 Activity= 70 QOL= 49.1
kno(i,1)= 7 kno(i,2)= 6 kno(i,3)= 11 kno(i,knomax)= 4
i= 7 x= 16 y= 6 Hunger= 100 Fatigue= 0 Companionship= 100 Activity= 25 QOL= 19
kno(i,1)= 7 kno(i,2)= 16 kno(i,3)= 11 kno(i,knomax)= 3
i= 8 x= 11 y= 6 Hunger= 100 Fatigue= 16 Companionship= 100 Activity= 9 QOL= 23
kno(i,1)= 7 kno(i,2)= 11 kno(i,3)= 6 kno(i,knomax)= 1
i= 9 x= 8 y= 19 Hunger= 76 Fatigue= 67 Companionship= 77 Activity= 9 QOL= 77.6
kno(i,1)= 7 kno(i,2)= 8 kno(i,3)= 19 kno(i,knomax)= 3
i= 10 x= 4 y= 12 Hunger= 69 Fatigue= 80 Companionship= 70 Activity= 79 QOL= 104.4
kno(i,1)= 2 kno(i,2)= 6 kno(i,3)= 11 kno(i,knomax)= 4
i= 11 x= 1 y= 19 Hunger= 100 Fatigue= 0 Companionship= 80 Activity= 25 QOL= 85
kno(i,1)= 7 kno(i,2)= 3 kno(i,3)= 9 kno(i,knomax)= 1

i= 12 x= 11 y= 15 Hunger= 100 Fatigue= 0 Companionship= 100 Activity= 21 QOL= 17
    kno(i,1)= 7 kno(i,2)= 11 kno(i,3)= 16 kno(i,knomax)= 2
i= 13 x= 9 y= 12 Hunger= 100 Fatigue= 0 Companionship= 100 Activity= 25 QOL= 9
    kno(i,1)= 7 kno(i,2)= kno(i,3)= kno(i,knomax)=
i= 14 x= 5 y= 4 Hunger= 100 Fatigue= 0 Companionship= 100 Activity= 25 QOL= 9
    kno(i,1)= 7 kno(i,2)= kno(i,3)= kno(i,knomax)=
i= 15 x= 19 y= 17 Hunger= 100 Fatigue= 0 Companionship= 100 Activity= 25 QOL= 9
    kno(i,1)= 7 kno(i,2)= kno(i,3)= kno(i,knomax)=
i= 16 x= 4 y= 12 Hunger= 70 Fatigue= 0 Companionship= 70 Activity= 71 QOL= 49.1
    kno(i,1)= 7 kno(i,2)= 6 kno(i,3)= 11 kno(i,knomax)= 4
i= 17 x= 13 y= 8 Hunger= 100 Fatigue= 0 Companionship= 80 Activity= 25 QOL= 91
    kno(i,1)= 7 kno(i,2)= 19 kno(i,3)= 13 kno(i,knomax)= 3
i= 18 x= 11 y= 6 Hunger= 100 Fatigue= 16 Companionship= 100 Activity= 9 QOL= 23
    kno(i,1)= 7 kno(i,2)= 11 kno(i,3)= 6 kno(i,knomax)= 1
i= 19 x= 8 y= 19 Hunger= 76 Fatigue= 78 Companionship= 77 Activity= 77 QOL= 89.2
    kno(i,1)= 2 kno(i,2)= 13 kno(i,3)= 19 kno(i,knomax)= 3
i= 20 x= 6 y= 12 Hunger= 100 Fatigue= 0 Companionship= 100 Activity= 21 QOL= 17
    kno(i,1)= 7 kno(i,2)= 6 kno(i,3)= 11 kno(i,knomax)= 2
Agent 1 communicated information to agent 7 at t= 101
Agent 1 shared 200 with agent 7 at t= 101
Agent 1 communicated information to agent 4 at t= 112
Agent 1 communicated information to agent 14 at t= 112
Agent 11 updated information to agent 1 at t= 120
Agent 11 communicated information to agent 2 at t= 120
Agent 11 communicated information to agent 12 at t= 120
Agent 11 communicated information to agent 18 at t= 120
Agent 1 communicated information to agent 10 at t= 134
Agent 1 updated information to agent 10 at t= 134
Agent 10 communicated information to agent 1 at t= 134
Agent 17 stole 400 from agent 4 at t= 136
Food Center 3 was emptied at t= 136
Food Center 3 was emptied at t= 140
Agent 20 stole 400 from agent 14 at t= 141
Revolution at shelter 8 with agent 14 replacing agent 20 at time= 142
Agent 3 communicated information to agent 7 at t= 143
Food Center 3 was emptied at t= 157
Agent 19 communicated information to agent 1 at t= 158
Food Center 3 was emptied at t= 159
Food Center 3 was emptied at t= 162
Agent 3 updated information to agent 7 at t= 173
Agent 10 communicated information to agent 7 at t= 179
Agent 10 updated information to agent 7 at t= 179
Agent 10 communicated information to agent 3 at t= 181
Agent 10 updated information to agent 3 at t= 181
Agent 3 updated information to agent 10 at t= 182

Later in the run…

t= 3000 Population= 67 Births= 47 Deaths = 0

Number of good agents= 31
Average quality of life of good agents this timestep= 1854.89032258065
Average strength of good agents this timestep= 1015.03225806452
Average power factor of good agents this timestep= 27103.5814265749
Average social factor of good agents this timestep= 26332.4552975427

Number of bad agents= 36
Average quality of life of bad agents this timestep= 1429.72222222222
Average strength of bad agents this timestep= 1002.27777777778
Average power factor of bad agents this timestep= 25534.220425257
Average social factor of bad agents this timestep= 25126.7926474792

Shelters

3	9	0	0	0		
16	11	0	0	0		
6	6	155992	35998	19	38	
9	3	0	0	0		
19	13	0	0	0		
13	19	0	0	0		
11	6	0	0	0		
16	16	175663	32327	14	29	
6	11	0	0	0		
11	16	0	0	0		

Food Centers

3	3	13369	20
6	16	36961	20
11	11	4065	20
16	6	34599	20
19	19	30529	20

Material Centers

11	3	981600
3	11	844400
19	11	919200
11	19	890400
3	19	999600
19	3	1000000

Agents
i= 1 x= 6 y= 1 Hunger= 66 Fatigue= 67 Companionship= 50 Activity= 65 QOL= 4619.6
kno(i,1)= 2 kno(i,2)= 6 kno(i,3)= 6 kno(i,knomax)= 21
i= 2 x= 6 y= 6 Hunger= 30 Fatigue= 0 Companionship= 3 Activity= 3 QOL= 2734.1
kno(i,1)= 2 kno(i,2)= 6 kno(i,3)= 6 kno(i,knomax)= 21
i= 3 x= 20 y= 16 Hunger= 99 Fatigue= 0 Companionship= 97 Activity= 99 QOL= 4282.5
kno(i,1)= 7 kno(i,2)= 16 kno(i,3)= 16 kno(i,knomax)= 21
i= 4 x= 20 y= 16 Hunger= 99 Fatigue= 0 Companionship= 95 Activity= 98 QOL= 2994.9
kno(i,1)= 7 kno(i,2)= 16 kno(i,3)= 16 kno(i,knomax)= 21
i= 5 x= 6 y= 6 Hunger= 12 Fatigue= 0 Companionship= 3 Activity= 11 QOL= 2742
kno(i,1)= 2 kno(i,2)= 6 kno(i,3)= 6 kno(i,knomax)= 21
i= 6 x= 11 y= 18 Hunger= 7 Fatigue= 8 Companionship= 34 Activity= 8 QOL= 2278.9
kno(i,1)= 7 kno(i,2)= 11 kno(i,3)= 18 kno(i,knomax)= 21
i= 7 x= 11 y= 19 Hunger= 94 Fatigue= 0 Companionship= 86 Activity= 94 QOL= 3953.4
kno(i,1)= 2 kno(i,2)= 13 kno(i,3)= 19 kno(i,knomax)= 21
i= 8 x= 6 y= 6 Hunger= 3 Fatigue= 0 Companionship= 3 Activity= 11 QOL= 2405.1
kno(i,1)= 2 kno(i,2)= 3 kno(i,3)= 3 kno(i,knomax)= 21
i= 9 x= 3 y= 10 Hunger= 28 Fatigue= 29 Companionship= 13 Activity= 33 QOL= 1341.7
kno(i,1)=-1 kno(i,2)= 6 kno(i,3)= 6 kno(i,knomax)= 21
i= 10 x= 16 y= 12 Hunger= 51 Fatigue= 53 Companionship= 42 Activity= 52 QOL= 5377.2

kno(i,1)= 2 kno(i,2)= 16 kno(i,3)= 11 kno(i,knomax)= 21
i= 11 x= 10 y= 7 Hunger= 77 Fatigue= 70 Companionship= 68 Activity= 8 QOL= 3331.4
kno(i,1)= 1 kno(i,2)= 11 kno(i,3)= 11 kno(i,knomax)= 21
i= 12 x= 3 y= 11 Hunger= 22 Fatigue= 23 Companionship= 21 Activity= 22 QOL= 3033.9
kno(i,1)= 2 kno(i,2)= 3 kno(i,3)= 9 kno(i,knomax)= 21
i= 13 x= 11 y= 19 Hunger= 25 Fatigue= 60 Companionship= 105 Activity= 0 QOL= 2931.4
kno(i,1)=-6 kno(i,2)= 11 kno(i,3)= 19 kno(i,knomax)= 21
i= 14 x= 1 y= 5 Hunger= 87 Fatigue= 0 Companionship= 63 Activity= 98 QOL= 449.5
kno(i,1)= 7 kno(i,2)= 6 kno(i,3)= 6 kno(i,knomax)= 21
i= 15 x= 6 y= 7 Hunger= 37 Fatigue= 30 Companionship= 1 Activity= 47 QOL= 3037.2
kno(i,1)=-6 kno(i,2)= 3 kno(i,3)= 11 kno(i,knomax)= 21
i= 16 x= 11 y= 19 Hunger= 94 Fatigue= 0 Companionship= 86 Activity= 94 QOL= 2933.7
kno(i,1)= 2 kno(i,2)= 13 kno(i,3)= 19 kno(i,knomax)= 21
i= 17 x= 16 y= 12 Hunger= 68 Fatigue= 72 Companionship= 44 Activity= 49 QOL= 3344.8
kno(i,1)= 2 kno(i,2)= 16 kno(i,3)= 11 kno(i,knomax)= 21
i= 18 x= 10 y= 6 Hunger= 70 Fatigue= 71 Companionship= 70 Activity= 70 QOL= 2737.6
kno(i,1)= 2 kno(i,2)= 11 kno(i,3)= 6 kno(i,knomax)= 21
i= 19 x= 2 y= 5 Hunger= 93 Fatigue= 62 Companionship= 25 Activity= 66 QOL= 2600.8
kno(i,1)= 1 kno(i,2)= 3 kno(i,3)= 3 kno(i,knomax)= 21
i= 20 x= 19 y= 11 Hunger= 45 Fatigue= 46 Companionship= 43 Activity= 45 QOL= 3473
kno(i,1)= 2 kno(i,2)= 19 kno(i,3)= 13 kno(i,knomax)= 21
i= 21 x= 15 y= 18 Hunger= 14 Fatigue= 0 Companionship= 40 Activity= 107 QOL= 1996.6
kno(i,1)= 2 kno(i,2)= 13 kno(i,3)= 19 kno(i,knomax)= 21
i= 22 x= 19 y= 11 Hunger= 43 Fatigue= 44 Companionship= 41 Activity= 43 QOL= 2740.1
kno(i,1)= 2 kno(i,2)= 19 kno(i,3)= 13 kno(i,knomax)= 21
i= 23 x= 2 y= 6 Hunger= 70 Fatigue= 71 Companionship= 69 Activity= 70 QOL= 1339.3
kno(i,1)= 2 kno(i,2)= 6 kno(i,3)= 6 kno(i,knomax)= 21
i= 24 x= 3 y= 11 Hunger= 20 Fatigue= 21 Companionship= 20 Activity= 20 QOL= 1041.8
kno(i,1)= 2 kno(i,2)= 3 kno(i,3)= 9 kno(i,knomax)= 21
i= 25 x= 19 y= 11 Hunger= 41 Fatigue= 42 Companionship= 40 Activity= 41 QOL= 2906.6
kno(i,1)= 2 kno(i,2)= 19 kno(i,3)= 13 kno(i,knomax)= 21
i= 26 x= 3 y= 11 Hunger= 20 Fatigue= 21 Companionship= 20 Activity= 20 QOL= 1564
kno(i,1)= 2 kno(i,2)= 3 kno(i,3)= 9 kno(i,knomax)= 21
i= 27 x= 6 y= 6 Hunger= 45 Fatigue= 0 Companionship= 3 Activity= 3 QOL= 1564
kno(i,1)= 2 kno(i,2)= 6 kno(i,3)= 6 kno(i,knomax)= 21
i= 28 x= 6 y= 6 Hunger= 11 Fatigue= 19 Companionship= 0 Activity= 25 QOL= 1929.2
kno(i,1)= 8 kno(i,2)= 6 kno(i,3)= 6 kno(i,knomax)= 21
i= 29 x= 16 y= 16 Hunger= 27 Fatigue= 20 Companionship= 28 Activity= 0 QOL= 1870.6
kno(i,1)= 7 kno(i,2)= 16 kno(i,3)= 16 kno(i,knomax)= 21
i= 30 x= 6 y= 2 Hunger= 54 Fatigue= 55 Companionship= 51 Activity= 54 QOL= 1778.8
kno(i,1)= 2 kno(i,2)= 6 kno(i,3)= 6 kno(i,knomax)= 21
i= 31 x= 16 y= 12 Hunger= 50 Fatigue= 51 Companionship= 45 Activity= 50 QOL= 1756.2
kno(i,1)= 2 kno(i,2)= 16 kno(i,3)= 11 kno(i,knomax)= 21
i= 32 x= 21 y= 10 Hunger= 55 Fatigue= 60 Companionship= 30 Activity= 68 QOL= 1411.2
kno(i,1)= 8 kno(i,2)= 21 kno(i,3)= 10 kno(i,knomax)= 21
i= 33 x= 4 y= 1 Hunger= 101 Fatigue= 3 Companionship= 2 Activity= 98 QOL= 1358.3
kno(i,1)= 1 kno(i,2)= 3 kno(i,3)= 3 kno(i,knomax)= 21
i= 34 x= 2 y= 5 Hunger= 79 Fatigue= 71 Companionship= 65 Activity= 12 QOL= 1453.3
kno(i,1)= 1 kno(i,2)= 3 kno(i,3)= 3 kno(i,knomax)= 21
i= 35 x= 15 y= 17 Hunger= 33 Fatigue= 34 Companionship= 43 Activity= 8 QOL= 1419.2
kno(i,1)=-1 kno(i,2)= 16 kno(i,3)= 16 kno(i,knomax)= 21
i= 36 x= 4 y= 2 Hunger= 134 Fatigue= 36 Companionship= 60 Activity= 100 QOL= 1140.2
kno(i,1)= 1 kno(i,2)= 6 kno(i,3)= 6 kno(i,knomax)= 21
i= 37 x= 19 y= 11 Hunger= 42 Fatigue= 43 Companionship= 41 Activity= 42 QOL= 1891.9
kno(i,1)= 2 kno(i,2)= 19 kno(i,3)= 13 kno(i,knomax)= 21
i= 38 x= 20 y= 10 Hunger= 54 Fatigue= 75 Companionship= 158 Activity= 83 QOL=-354.7

kno(i,1)=-1 kno(i,2)= 6 kno(i,3)= 6 kno(i,knomax)= 21
i= 39 x= 19 y= 12 Hunger= 42 Fatigue= 43 Companionship= 49 Activity= 8 QOL= 1451.6
kno(i,1)=-1 kno(i,2)= 16 kno(i,3)= 16 kno(i,knomax)= 21
i= 40 x= 2 y= 5 Hunger= 58 Fatigue= 60 Companionship= 145 Activity= 92 QOL= 849.9
kno(i,1)= 7 kno(i,2)= 2 kno(i,3)= 5 kno(i,knomax)= 21
i= 41 x= 6 y= 6 Hunger= 22 Fatigue= 23 Companionship= 29 Activity= 0 QOL= 1123.3
kno(i,1)= 7 kno(i,2)= 6 kno(i,3)= 6 kno(i,knomax)= 21
i= 42 x= 11 y= 19 Hunger= 94 Fatigue= 0 Companionship= 84 Activity= 94 QOL= 1074.8
kno(i,1)= 2 kno(i,2)= 13 kno(i,3)= 19 kno(i,knomax)= 21
i= 43 x= 3 y= 6 Hunger= 52 Fatigue= 55 Companionship= 52 Activity= 54 QOL= 1005.1
kno(i,1)= 2 kno(i,2)= 6 kno(i,3)= 6 kno(i,knomax)= 21
i= 44 x= 2 y= 6 Hunger= 66 Fatigue= 67 Companionship= 67 Activity= 4 QOL= 505.9
kno(i,1)= 7 kno(i,2)= 2 kno(i,3)= 6 kno(i,knomax)= 21
i= 45 x= 10 y= 6 Hunger= 67 Fatigue= 68 Companionship= 67 Activity= 66 QOL= 1139.4
kno(i,1)= 2 kno(i,2)= 11 kno(i,3)= 6 kno(i,knomax)= 21
i= 46 x= 19 y= 11 Hunger= 45 Fatigue= 46 Companionship= 42 Activity= 45 QOL= 1660.7
kno(i,1)= 2 kno(i,2)= 19 kno(i,3)= 13 kno(i,knomax)= 21
i= 47 x= 10 y= 6 Hunger= 68 Fatigue= 69 Companionship= 65 Activity= 68 QOL= 418.5
kno(i,1)= 2 kno(i,2)= 11 kno(i,3)= 6 kno(i,knomax)= 21
i= 48 x= 6 y= 8 Hunger= 4 Fatigue= 5 Companionship= 0 Activity= 8 QOL= 996.5
kno(i,1)= 8 kno(i,2)= 6 kno(i,3)= 8 kno(i,knomax)= 21
i= 49 x= 11 y= 19 Hunger= 93 Fatigue= 0 Companionship= 84 Activity= 93 QOL= 1244.6
kno(i,1)= 2 kno(i,2)= 13 kno(i,3)= 19 kno(i,knomax)= 21
i= 50 x= 1 y= 6 Hunger= 80 Fatigue= 89 Companionship= 62 Activity= 87 QOL= 697.7
kno(i,1)= 2 kno(i,2)= 6 kno(i,3)= 6 kno(i,knomax)= 21
i= 51 x= 1 y= 6 Hunger= 80 Fatigue= 88 Companionship= 44 Activity= 86 QOL= 537.2
kno(i,1)= 2 kno(i,2)= 6 kno(i,3)= 6 kno(i,knomax)= 21
i= 52 x= 1 y= 1 Hunger= 93 Fatigue= 0 Companionship= 44 Activity= 96 QOL= 917.7
kno(i,1)= 2 kno(i,2)= 6 kno(i,3)= 6 kno(i,knomax)= 21
i= 53 x= 16 y= 14 Hunger= 65 Fatigue= 66 Companionship= 43 Activity= 46 QOL= 1457.3
kno(i,1)= 2 kno(i,2)= 16 kno(i,3)= 16 kno(i,knomax)= 21
i= 54 x= 3 y= 11 Hunger= 66 Fatigue= 67 Companionship= 62 Activity= 66 QOL= 641.5
kno(i,1)= 2 kno(i,2)= 3 kno(i,3)= 9 kno(i,knomax)= 21
i= 55 x= 6 y= 4 Hunger= 56 Fatigue= 0 Companionship= 57 Activity= 88 QOL= 776.1
kno(i,1)= 2 kno(i,2)= 6 kno(i,3)= 6 kno(i,knomax)= 21
i= 56 x= 9 y= 6 Hunger= 64 Fatigue= 71 Companionship= 63 Activity= 70 QOL= 1270.5
kno(i,1)= 2 kno(i,2)= 6 kno(i,3)= 6 kno(i,knomax)= 21
i= 57 x= 16 y= 16 Hunger= 0 Fatigue= 20 Companionship= 2 Activity= 0 QOL= 922.3
kno(i,1)= 1 kno(i,2)= 19 kno(i,3)= 19 kno(i,knomax)= 21
i= 58 x= 1 y= 20 Hunger= 53 Fatigue= 54 Companionship= 72 Activity= 92 QOL=-195.7
kno(i,1)= 8 kno(i,2)= 1 kno(i,3)= 20 kno(i,knomax)= 21
i= 59 x= 19 y= 6 Hunger= 50 Fatigue= 55 Companionship= 119 Activity= 80 QOL= 118.5
kno(i,1)= 7 kno(i,2)= 19 kno(i,3)= 6 kno(i,knomax)= 21
i= 60 x= 3 y= 10 Hunger= 73 Fatigue= 22 Companionship= 62 Activity= 81 QOL= 136.1
kno(i,1)=-1 kno(i,2)= 6 kno(i,3)= 6 kno(i,knomax)= 21
i= 61 x= 15 y= 18 Hunger= 82 Fatigue= 0 Companionship= 39 Activity= 86 QOL= 300.8
kno(i,1)= 2 kno(i,2)= 13 kno(i,3)= 19 kno(i,knomax)= 21
i= 62 x= 5 y= 2 Hunger= 69 Fatigue= 42 Companionship= 60 Activity= 71 QOL= 46
kno(i,1)= 8 kno(i,2)= 5 kno(i,3)= 2 kno(i,knomax)= 21
i= 63 x= 2 y= 4 Hunger= 79 Fatigue= 72 Companionship= 65 Activity= 12 QOL= 235
kno(i,1)= 1 kno(i,2)= 3 kno(i,3)= 3 kno(i,knomax)= 21
i= 64 x= 19 y= 10 Hunger= 56 Fatigue= 66 Companionship= 166 Activity= 25 QOL=-57
kno(i,1)=-1 kno(i,2)= 6 kno(i,3)= 6 kno(i,knomax)= 21
i= 65 x= 3 y= 11 Hunger= 27 Fatigue= 17 Companionship= 27 Activity= 17 QOL= 176
kno(i,1)= 7 kno(i,2)= 3 kno(i,3)= 11 kno(i,knomax)= 21
i= 66 x= 19 y= 13 Hunger= 49 Fatigue= 42 Companionship= 49 Activity= 7 QOL= 77

kno(i,1)=-1 kno(i,2)= 16 kno(i,3)= 16 kno(i,knomax)= 21
i= 67 x= 3 y= 2 Hunger= 34 Fatigue= 34 Companionship= 33 Activity=-1 QOL= 35
kno(i,1)= 1 kno(i,2)= 3 kno(i,3)= 3 kno(i,knomax)= 21
Agent 6 stole 13 from agent 7 at t= 3002
Agent 52 updated information to agent 14 at t= 3004
Agent 42 updated information to agent 6 at t= 3006
Agent 61 updated information to agent 21 at t= 3006
Agent 61 updated information to agent 35 at t= 3006
Agent 21 updated information to agent 61 at t= 3007
Agent 49 updated information to agent 6 at t= 3007
Agent 34 updated information to agent 40 at t= 3009
Agent 68 born to parents 34 and 23 at t= 3009
1 2 6 0 0 0 0 0 0 0 0 3 0 0.9442123 0 0 1 252 0.3612987 0 452 0 34 200
Agent 61 updated information to agent 21 at t= 3009
Agent 21 updated information to agent 35 at t= 3010
Agent 21 updated information to agent 57 at t= 3012
Agent 67 updated information to agent 41 at t= 3016
Agent 67 shared 200 with agent 36 at t= 3016
Agent 16 stole 400 from agent 7 at t= 3017
Agent 67 shared 100 with agent 33 at t= 3017
Agent 33 updated information to agent 67 at t= 3020
Agent 50 updated information to agent 55 at t= 3020
Agent 50 shared 30 with agent 55 at t= 3020
Agent 67 updated information to agent 52 at t= 3020
Agent 3 updated information to agent 4 at t= 3021
Agent 3 updated information to agent 29 at t= 3021
Agent 69 born to parents 21 and 35 at t= 3023
1 13 18 0 0 0 0 0 0 0 0 8 1 0.6223674 0 0 0 696 0.4616345 0 896 0 21 200
Agent 67 updated information to agent 30 at t= 3023
Agent 67 updated information to agent 36 at t= 3023
Agent 67 shared 41 with agent 30 at t= 3023
Agents 36 and 33 mated at t= 3025
Agents 21 and 35 mated at t= 3027
Agents 23 and 34 mated at t= 3030
Agent 56 updated information to agent 47 at t= 3031
Agent 15 updated information to agent 9 at t= 3032
Agent 15 updated information to agent 12 at t= 3032
Agent 15 updated information to agent 24 at t= 3032
Agent 15 updated information to agent 26 at t= 3032
Agent 15 updated information to agent 65 at t= 3032
Agent 47 updated information to agent 45 at t= 3032
Agent 47 updated information to agent 56 at t= 3032
Agent 21 updated information to agent 29 at t= 3033
Agent 56 updated information to agent 47 at t= 3033
Agent 1 updated information to agent 16 at t= 3036
Agents 50 and 51 mated at t= 3036
Agent 21 updated information to agent 42 at t= 3037
Agent 21 updated information to agent 49 at t= 3038
Agent 15 updated information to agent 41 at t= 3040
Agent 63 updated information to agent 8 at t= 3040
Agent 15 updated information to agent 34 at t= 3041
Agent 15 updated information to agent 63 at t= 3041
Agent 70 born to parents 38 and 32 at t= 3041
1 18 10 0 0 0 0 0 0 0 0 3 1 0.1517575 0 0 1 492 0.3744007 0 692 0 38 200
Agent 2 stole 400 from agent 68 at t= 3042
Agent 15 updated information to agent 23 at t= 3042

Agent 27 stole 400 from agent 2 at t= 3042
Agents 38 and 32 mated at t= 3042
Agent 70 updated information to agent 64 at t= 3042
Agent 15 updated information to agent 44 at t= 3045
Agent 15 updated information to agent 11 at t= 3046
Agent 30 updated information to agent 62 at t= 3046
Agent 30 shared 20 with agent 62 at t= 3046
Agent 15 updated information to agent 18 at t= 3047
Agent 15 updated information to agent 68 at t= 3047
Agent 68 updated information to agent 5 at t= 3047
Agent 68 updated information to agent 15 at t= 3047
Agent 68 updated information to agent 34 at t= 3047
Agent 68 updated information to agent 63 at t= 3047
Agent 15 updated information to agent 8 at t= 3048
Agent 68 updated information to agent 23 at t= 3048
Agent 21 updated information to agent 52 at t= 3054
Agent 30 updated information to agent 51 at t= 3054
Agent 50 shared 15 with agent 30 at t= 3054
Agent 20 stole 400 from agent 66 at t= 3055
Agent 46 stole 400 from agent 20 at t= 3055
Food Center 3 was emptied at t= 3056
Food Center 3 was emptied at t= 3057
Agent 37 stole 400 from agent 39 at t= 3058
Agents 39 and 25 mated at t= 3058
Agent 55 updated information to agent 29 at t= 3058
Agent 55 updated information to agent 42 at t= 3058
Agent 55 updated information to agent 49 at t= 3058
Agent 55 updated information to agent 57 at t= 3058
Agent 25 updated information to agent 37 at t= 3059
Agent 25 updated information to agent 39 at t= 3059
Agent 25 updated information to agent 66 at t= 3059
Agent 54 updated information to agent 41 at t= 3061
Agents 54 and 41 mated at t= 3061
Agent 40 stole 400 from agent 10 at t= 3062
Agent 43 stole 400 from agent 41 at t= 3064
Agent 54 updated information to agent 30 at t= 3064
Agent 54 updated information to agent 43 at t= 3064
Agent 54 updated information to agent 59 at t= 3064
Food Center 3 was emptied at t= 3064
Agent 54 updated information to agent 34 at t= 3065
Agent 54 updated information to agent 63 at t= 3065
Agent 54 updated information to agent 8 at t= 3066
Agent 54 updated information to agent 23 at t= 3066
Agent 54 updated information to agent 68 at t= 3066
Food Center 3 was emptied at t= 3066
Agent 54 updated information to agent 2 at t= 3067
Agent 54 updated information to agent 27 at t= 3067
Food Center 3 was emptied at t= 3067
Food Center 3 was emptied at t= 3068
Agent 25 updated information to agent 29 at t= 3069
Agent 25 updated information to agent 42 at t= 3069
Agent 25 updated information to agent 49 at t= 3069
Agent 25 updated information to agent 57 at t= 3069
Agent 54 updated information to agent 41 at t= 3069
Food Center 3 was emptied at t= 3069
Food Center 3 was emptied at t= 3070

Agent 53 updated information to agent 10 at t= 3071
Agent 53 updated information to agent 17 at t= 3071
Agent 11 updated information to agent 5 at t= 3072
Agent 11 updated information to agent 15 at t= 3072
Agent 11 updated information to agent 30 at t= 3072
Agent 11 updated information to agent 43 at t= 3072
Agent 11 updated information to agent 44 at t= 3072
Agent 11 updated information to agent 59 at t= 3072
Agent 11 updated information to agent 63 at t= 3072
Agent 30 updated information to agent 5 at t= 3072
Agent 30 updated information to agent 11 at t= 3072
Agent 30 updated information to agent 15 at t= 3072
Agent 30 updated information to agent 18 at t= 3072
Agent 30 updated information to agent 43 at t= 3072
Agent 30 updated information to agent 44 at t= 3072
Agent 30 updated information to agent 59 at t= 3072
Agent 30 updated information to agent 63 at t= 3072
Agent 71 born to parents 28 and 48 at t= 3073
1 6 10 0 0 0 0 0 0 0 0 0 3 1 0.6888914 0 0 1 484 0.3046458 0 684 0 28 200
Food Center 3 was emptied at t= 3073
Agent 55 updated information to agent 21 at t= 3074
Agent 55 updated information to agent 29 at t= 3074
Agent 55 updated information to agent 35 at t= 3074
Agent 55 updated information to agent 42 at t= 3074
Agent 55 updated information to agent 49 at t= 3074
Agent 55 updated information to agent 57 at t= 3074
Food Center 3 was emptied at t= 3074
Agent 19 updated information to agent 32 at t= 3075
Agent 55 updated information to agent 61 at t= 3075
Agent 55 updated information to agent 69 at t= 3075
Food Center 3 was emptied at t= 3075
Agent 19 updated information to agent 38 at t= 3076
Agent 21 updated information to agent 55 at t= 3076
Food Center 3 was emptied at t= 3076
Food Center 3 was emptied at t= 3077
Agent 21 updated information to agent 35 at t= 3078
Food Center 3 was emptied at t= 3078
Food Center 3 was emptied at t= 3080
Agent 35 updated information to agent 21 at t= 3081
Agent 35 updated information to agent 61 at t= 3082
Food Center 3 was emptied at t= 3082
Agents 28 and 48 mated at t= 3083
Food Center 3 was emptied at t= 3083
Food Center 3 was emptied at t= 3084
Agent 25 updated information to agent 29 at t= 3085
Agent 25 updated information to agent 30 at t= 3085
Agent 25 updated information to agent 39 at t= 3085
Agent 25 updated information to agent 42 at t= 3085
Agent 25 updated information to agent 49 at t= 3085
Agent 25 updated information to agent 57 at t= 3085
Food Center 3 was emptied at t= 3085
Food Center 3 was emptied at t= 3086
Agent 37 stole 400 from agent 40 at t= 3087
Agent 31 updated information to agent 17 at t= 3091
Agent 34 updated information to agent 1 at t= 3091
Agent 34 updated information to agent 8 at t= 3091

Agent 34 updated information to agent 45 at t= 3091
Agent 34 updated information to agent 47 at t= 3091
Agent 34 updated information to agent 56 at t= 3091
Agent 34 updated information to agent 68 at t= 3091
Agent 55 updated information to agent 13 at t= 3092
Agent 22 updated information to agent 31 at t= 3093
Agent 22 updated information to agent 46 at t= 3093
Agent 55 updated information to agent 6 at t= 3093
Agent 63 updated information to agent 2 at t= 3093
Agent 63 updated information to agent 9 at t= 3093
Agent 63 updated information to agent 12 at t= 3093
Agent 63 updated information to agent 23 at t= 3093
Agent 63 updated information to agent 27 at t= 3093
Agent 21 updated information to agent 69 at t= 3095
Agent 55 updated information to agent 52 at t= 3095
Agent 63 updated information to agent 15 at t= 3096
Agent 63 updated information to agent 44 at t= 3096
Agent 53 updated information to agent 10 at t= 3097
Food Center 3 was emptied at t= 3097
Agent 10 updated information to agent 53 at t= 3098
Food Center 3 was emptied at t= 3098
Agent 10 updated information to agent 17 at t= 3099
Agent 10 updated information to agent 20 at t= 3099
Agent 25 updated information to agent 29 at t= 3099
Agent 25 updated information to agent 30 at t= 3099
Agent 25 updated information to agent 39 at t= 3099
Agent 25 updated information to agent 42 at t= 3099
Agent 25 updated information to agent 49 at t= 3099
Agent 10 updated information to agent 31 at t= 3100
Agent 31 updated information to agent 10 at t= 3100
Agent 31 updated information to agent 22 at t= 3100

Table 6. File microssummary For Run With Input File Given in Table 4.

Micros Social Simulation

Run Prototype Civilization Scenario with NEW 21x21 landscape

Date 09-07-2002 Time 16:00:18

Size of Landscape= 21
Maxpop= 200
Number of agents= 20
Number of Food Centers= 5
Number of Material Centers= 6
Number of shelters= 10
Actor Lifetime= 4000
Length of Day= 24
Max food= 200
Max Sleep= 100
Max Sense Range = 2
Max Carry= 400
Shelter Deterioration Rate= 1
Workday= 12
Stoptime= 40000
Goodness parameter= 0.5
Leadership allowed=true
dtdump= 100
Gestation period= 200
Monogamy = true
Mate with only those of same shelter false

Shelters

3	9	0	0	0		0
16	11	0	0	0		0
6	6	100	0	2	10	0
9	3	0	0	0		0
19	13	0	0	0		0
13	19	0	0	0		0
11	6	0	0	0		0
16	16	100	0	3	10	0
6	11	0	0	0		0
11	16	0	0	0		0

Food Centers

3	3	100	20
6	16	100	20
11	11	100	20
16	6	100	20
19	19	100	20

Material Centers

11	3	1000000
3	11	1000000
19	11	1000000
11	19	1000000
3	19	1000000
19	3	1000000

Agents

0 2 8 100 0 0 0 1 0 0 0 3 1 0.2741053 1 224 0.9777883
0 7 15 100 0 0 0 1 0 0 0 3 1 0.4176249 0 104 0.8363596
0 9 12 100 0 0 0 1 0 0 0 8 1 0.2468173 1 292 0.2175482
0 5 3 100 0 0 0 1 0 0 0 8 0 0.9436283 1 600 0.8838664
0 18 17 100 0 0 0 1 0 0 0 3 1 0.4956676 0 508 0.8488029
0 4 12 100 0 0 0 1 0 0 0 8 1 0.1493344 0 476 0.454948
0 19 9 100 0 0 0 1 0 0 0 8 0 0.1129414 0 256 0.7021187
0 12 7 100 0 0 0 1 0 0 0 3 0 0.5098409 0 596 0.8254834
0 18 20 100 0 0 0 1 0 0 0 3 0 0.5815489 0 368 0.1236871
0 10 11 100 0 0 0 1 0 0 0 8 1 0.1408705 1 276 0.0369764
0 2 8 100 0 0 0 1 0 0 0 3 0 0.2750251 1 356 0.4753245
0 7 15 100 0 0 0 1 0 0 0 3 1 0.2987708 0 104 0.396556
0 9 12 100 0 0 0 1 0 0 0 8 0 0.9575302 1 384 0.6344722
0 5 3 100 0 0 0 1 0 0 0 8 1 0.8805146 0 732 0.976976
0 18 17 100 0 0 0 1 0 0 0 3 0 0.2838494 1 76 0.4941967
0 4 12 100 0 0 0 1 0 0 0 8 1 0.9236993 0 60 0.1166154
0 19 9 100 0 0 0 1 0 0 0 8 0 0.2993929 0 532 0.4631894
0 12 7 100 0 0 0 1 0 0 0 3 1 0.1065478 0 64 0.919478
0 18 20 100 0 0 0 1 0 0 0 3 0 0.9401957 1 696 0.9657667
0 10 11 100 0 0 0 1 0 0 0 8 0 0.2479076 0 788 0.7551928

Conditions at end of run

Shelters

3	9	0	0	0	
16	11	0	0	0	
6	6	908064	52995	144	66
9	3	0	0	0	
19	13	0	0	0	
13	19	0	0	0	
11	6	0	0	0	
16	16	928824	58968	135	44
6	11	0	0	0	
11	16	0	0	0	

Food Centers

3	3	0	20
6	16	1085	20
11	11	0	20
16	6	0	20
19	19	400	20

Material Centers

11	3	906800
3	11	20000
19	11	218800
11	19	697600
3	19	999600
19	3	998400

Agents
i= 91 x= 3 y= 17 Hunger= 27 Fatigue= 27 Companionship= 27 Activity=-1 QOL= 28
kno(i,1)= 7 kno(i,2)= 3 kno(i,3)= 17 kno(i,knomax)= 21
i= 92 x= 14 y= 6 Hunger= 31 Fatigue= 31 Companionship= 15 Activity=-1 QOL= 47
kno(i,1)= 1 kno(i,2)= 3 kno(i,3)= 3 kno(i,knomax)= 21
i= 93 x= 18 y= 11 Hunger= 23 Fatigue= 8 Companionship= 6 Activity= 15 QOL= 12
kno(i,1)= 1 kno(i,2)= 16 kno(i,3)= 6 kno(i,knomax)= 21
i= 94 x= 4 y= 10 Hunger= 101 Fatigue= 14 Companionship= 9 Activity= 92 QOL= 154.6
kno(i,1)= 1 kno(i,2)= 11 kno(i,3)= 11 kno(i,knomax)= 21
i= 95 x= 3 y= 4 Hunger= 13 Fatigue= 75 Companionship= 0 Activity= 38 QOL= 33
kno(i,1)= 2 kno(i,2)= 6 kno(i,3)= 6 kno(i,knomax)= 21
i= 96 x= 18 y= 13 Hunger= 20 Fatigue= 13 Companionship= 29 Activity= 16 QOL= 169.4
kno(i,1)=-1 kno(i,2)= 16 kno(i,3)= 16 kno(i,knomax)= 21
i= 97 x= 3 y= 3 Hunger= 40 Fatigue= 33 Companionship= 0 Activity= 29 QOL= 123
kno(i,1)= 1 kno(i,2)= 3 kno(i,3)= 3 kno(i,knomax)= 21
i= 98 x= 3 y= 4 Hunger= 85 Fatigue= 6 Companionship= 0 Activity= 98 QOL= 164.4
kno(i,1)= 8 kno(i,2)= 3 kno(i,3)= 4 kno(i,knomax)= 21
i= 99 x= 16 y= 6 Hunger= 32 Fatigue= 39 Companionship= 15 Activity= 35 QOL= 237.9
kno(i,1)= 2 kno(i,2)= 16 kno(i,3)= 11 kno(i,knomax)= 21
i= 100 x= 18 y= 11 Hunger= 103 Fatigue= 22 Companionship= 122 Activity= 47 QOL= 120
kno(i,1)= 1 kno(i,2)= 16 kno(i,3)= 6 kno(i,knomax)= 21
i= 101 x= 6 y= 5 Hunger= 4 Fatigue= 13 Companionship= 2 Activity= 13 QOL= 264.4
kno(i,1)= 8 kno(i,2)= 6 kno(i,3)= 5 kno(i,knomax)= 21
i= 102 x= 16 y= 6 Hunger= 43 Fatigue= 45 Companionship= 15 Activity= 44 QOL= 388.9
kno(i,1)= 2 kno(i,2)= 16 kno(i,3)= 11 kno(i,knomax)= 21
i= 103 x= 16 y= 6 Hunger= 30 Fatigue= 18 Companionship= 12 Activity= 81 QOL= 480.3
kno(i,1)= 1 kno(i,2)= 16 kno(i,3)= 6 kno(i,knomax)= 21
i= 104 x= 11 y= 11 Hunger= 79 Fatigue= 11 Companionship= 26 Activity= 63 QOL= 1190.8
kno(i,1)= 1 kno(i,2)= 11 kno(i,3)= 11 kno(i,knomax)= 21
i= 105 x= 4 y= 17 Hunger= 110 Fatigue= 61 Companionship= 58 Activity= 50 QOL= 468.4
kno(i,1)= 1 kno(i,2)= 6 kno(i,3)= 16 kno(i,knomax)= 21
i= 106 x= 16 y= 6 Hunger= 53 Fatigue= 49 Companionship= 14 Activity= 51 QOL= 1530.2
kno(i,1)= 1 kno(i,2)= 16 kno(i,3)= 6 kno(i,knomax)= 21
i= 107 x= 3 y= 3 Hunger= 86 Fatigue= 20 Companionship= 0 Activity= 104 QOL= 1372.1
kno(i,1)= 1 kno(i,2)= 3 kno(i,3)= 3 kno(i,knomax)= 21
i= 108 x= 16 y= 6 Hunger= 52 Fatigue= 55 Companionship= 15 Activity= 52 QOL= 1588.8
kno(i,1)= 2 kno(i,2)= 16 kno(i,3)= 11 kno(i,knomax)= 21
i= 109 x= 12 y= 16 Hunger= 19 Fatigue= 20 Companionship= 14 Activity= 19 QOL= 1606.1
kno(i,1)= 2 kno(i,2)= 11 kno(i,3)= 16 kno(i,knomax)= 21
i= 110 x= 3 y= 3 Hunger= 83 Fatigue= 4 Companionship= 1 Activity= 76 QOL= 1270.5
kno(i,1)= 1 kno(i,2)= 3 kno(i,3)= 3 kno(i,knomax)= 21
i= 111 x= 19 y= 13 Hunger= 11 Fatigue= 12 Companionship= 18 Activity= 8 QOL= 1310.7
kno(i,1)=-1 kno(i,2)= 16 kno(i,3)= 16 kno(i,knomax)= 21
i= 112 x= 4 y= 17 Hunger= 93 Fatigue= 14 Companionship= 65 Activity= 88 QOL= 1429.1
kno(i,1)= 1 kno(i,2)= 6 kno(i,3)= 16 kno(i,knomax)= 21
i= 113 x= 19 y= 12 Hunger= 14 Fatigue= 7 Companionship= 21 Activity= 8 QOL= 1292.1
kno(i,1)=-6 kno(i,2)= 19 kno(i,3)= 11 kno(i,knomax)= 21

i= 114 x= 4 y= 3 Hunger= 97 Fatigue= 18 Companionship= 2 Activity= 96 QOL= 1133.7
kno(i,1)= 8 kno(i,2)= 4 kno(i,3)= 3 kno(i,knomax)= 21
i= 115 x= 3 y= 10 Hunger= 51 Fatigue= 12 Companionship= 3 Activity= 82 QOL= 1078.7
kno(i,1)= 8 kno(i,2)= 3 kno(i,3)= 10 kno(i,knomax)= 21
i= 116 x= 3 y= 3 Hunger= 107 Fatigue= 8 Companionship= 2 Activity= 98 QOL= 2709.1
kno(i,1)= 1 kno(i,2)= 3 kno(i,3)= 3 kno(i,knomax)= 21
i= 117 x= 10 y= 6 Hunger= 44 Fatigue= 45 Companionship= 0 Activity= 44 QOL= 3189.1
kno(i,1)=-1 kno(i,2)= 6 kno(i,3)= 6 kno(i,knomax)= 21
i= 118 x= 12 y= 16 Hunger= 27 Fatigue= 28 Companionship= 25 Activity= 27 QOL= 2855.4
kno(i,1)= 2 kno(i,2)= 11 kno(i,3)= 16 kno(i,knomax)= 21
i= 119 x= 11 y= 11 Hunger= 33 Fatigue= 45 Companionship= 17 Activity= 42 QOL= 2296.3
kno(i,1)= 2 kno(i,2)= 6 kno(i,3)= 11 kno(i,knomax)= 21
i= 120 x= 14 y= 14 Hunger= 22 Fatigue= 49 Companionship= 16 Activity= 96 QOL= 2739.3
kno(i,1)=-1 kno(i,2)= 16 kno(i,3)= 16 kno(i,knomax)= 21
i= 121 x= 3 y= 3 Hunger= 84 Fatigue= 19 Companionship= 0 Activity= 99 QOL= 2980.7
kno(i,1)= 1 kno(i,2)= 3 kno(i,3)= 3 kno(i,knomax)= 21
i= 122 x= 6 y= 6 Hunger= 2 Fatigue= 3 Companionship= 1 Activity= 0 QOL= 2413.3
kno(i,1)= 8 kno(i,2)= 6 kno(i,3)= 6 kno(i,knomax)= 21
i= 123 x= 18 y= 12 Hunger= 6 Fatigue= 9 Companionship= 7 Activity= 8 QOL= 2569.1
kno(i,1)=-6 kno(i,2)= 19 kno(i,3)= 11 kno(i,knomax)= 21
i= 124 x= 19 y= 12 Hunger= 34 Fatigue= 66 Companionship= 109 Activity= 0 QOL= 1209.6
kno(i,1)=-6 kno(i,2)= 19 kno(i,3)= 11 kno(i,knomax)= 21
i= 125 x= 10 y= 9 Hunger= 102 Fatigue= 0 Companionship= 8 Activity= 28 QOL= 1887.5
kno(i,1)= 2 kno(i,2)= 6 kno(i,3)= 6 kno(i,knomax)= 21
i= 126 x= 15 y= 14 Hunger= 40 Fatigue= 49 Companionship= 24 Activity= 99 QOL= 1944.9
kno(i,1)=-1 kno(i,2)= 16 kno(i,3)= 16 kno(i,knomax)= 21
i= 127 x= 6 y= 15 Hunger= 1 Fatigue= 11 Companionship= 26 Activity= 62 QOL= 2467.4
kno(i,1)=-1 kno(i,2)= 6 kno(i,3)= 6 kno(i,knomax)= 21
i= 128 x= 10 y= 10 Hunger= 102 Fatigue= 5 Companionship= 40 Activity= 93 QOL= 2448.9
kno(i,1)= 1 kno(i,2)= 11 kno(i,3)= 11 kno(i,knomax)= 21
i= 129 x= 6 y= 6 Hunger= 6 Fatigue= 19 Companionship= 0 Activity= 99 QOL= 2487
kno(i,1)=-1 kno(i,2)= 16 kno(i,3)= 16 kno(i,knomax)= 21
i= 130 x= 7 y= 7 Hunger= 164 Fatigue= 48 Companionship= 2 Activity= 261 QOL= 851.4
kno(i,1)= 1 kno(i,2)= 16 kno(i,3)= 16 kno(i,knomax)= 21
i= 131 x= 2 y= 11 Hunger= 70 Fatigue= 11 Companionship= 71 Activity= 69 QOL= 2247.7
kno(i,1)= 8 kno(i,2)= 2 kno(i,3)= 11 kno(i,knomax)= 21
i= 132 x= 6 y= 6 Hunger= 0 Fatigue= 1 Companionship= 1 Activity= 8 QOL= 3914.7
kno(i,1)= 1 kno(i,2)= 6 kno(i,3)= 6 kno(i,knomax)= 21
i= 133 x= 3 y= 11 Hunger= 17 Fatigue= 18 Companionship= 16 Activity= 17 QOL= 3295.2
kno(i,1)= 2 kno(i,2)= 3 kno(i,3)= 9 kno(i,knomax)= 21
i= 134 x= 20 y= 2 Hunger= 81 Fatigue= 83 Companionship= 72 Activity= 82 QOL= 3336.7
kno(i,1)= 2 kno(i,2)= 19 kno(i,3)= 13 kno(i,knomax)= 21
i= 135 x= 3 y= 4 Hunger= 30 Fatigue= 20 Companionship= 0 Activity= 142 QOL= 4445
kno(i,1)= 8 kno(i,2)= 4 kno(i,3)= 3 kno(i,knomax)= 21
i= 136 x= 10 y= 1 Hunger= 60 Fatigue= 15 Companionship= 11 Activity= 71 QOL= 4598.5
kno(i,1)= 8 kno(i,2)= 10 kno(i,3)= 1 kno(i,knomax)= 21
i= 137 x= 20 y= 2 Hunger= 134 Fatigue= 81 Companionship= 57 Activity= 82 QOL= 4585.3
kno(i,1)= 2 kno(i,2)= 19 kno(i,3)= 13 kno(i,knomax)= 21
i= 138 x= 3 y= 5 Hunger= 121 Fatigue= 22 Companionship= 0 Activity= 98 QOL= 1711.8
kno(i,1)= 1 kno(i,2)= 6 kno(i,3)= 16 kno(i,knomax)= 21
i= 139 x= 6 y= 6 Hunger= 10 Fatigue= 3 Companionship= 0 Activity= 42 QOL= 2679.7
kno(i,1)= 7 kno(i,2)= 3 kno(i,3)= 3 kno(i,knomax)= 21
i= 140 x= 3 y= 4 Hunger= 32 Fatigue= 39 Companionship= 0 Activity= 63 QOL= 1870
kno(i,1)= 8 kno(i,2)= 3 kno(i,3)= 4 kno(i,knomax)= 21
i= 141 x= 6 y= 6 Hunger= 0 Fatigue= 2 Companionship= 47 Activity= 95 QOL= 3862.9
kno(i,1)= 1 kno(i,2)= 6 kno(i,3)= 16 kno(i,knomax)= 21

i= 142 x= 6 y= 2 Hunger= 25 Fatigue= 26 Companionship= 25 Activity= 25 QOL= 3199.1
kno(i,1)= 2 kno(i,2)= 6 kno(i,3)= 6 kno(i,knomax)= 21
i= 143 x= 3 y= 5 Hunger= 40 Fatigue= 33 Companionship= 0 Activity= 7 QOL= 3568.9
kno(i,1)= 1 kno(i,2)= 6 kno(i,3)= 16 kno(i,knomax)= 21
i= 144 x= 10 y= 2 Hunger= 101 Fatigue= 25 Companionship= 1 Activity= 104 QOL= 5028.7
kno(i,1)= 1 kno(i,2)= 11 kno(i,3)= 11 kno(i,knomax)= 21
i= 145 x= 4 y= 3 Hunger= 43 Fatigue= 46 Companionship= 16 Activity= 18 QOL= 89.3
kno(i,1)= 2 kno(i,2)= 6 kno(i,3)= 6 kno(i,knomax)= 21
i= 146 x= 11 y= 16 Hunger= 44 Fatigue= 22 Companionship= 11 Activity= 28 QOL= 123.4
kno(i,1)= 1 kno(i,2)= 6 kno(i,3)= 16 kno(i,knomax)= 21
i= 147 x= 17 y= 18 Hunger= 104 Fatigue= 24 Companionship= 15 Activity= 40 QOL= 88
kno(i,1)= 1 kno(i,2)= 19 kno(i,3)= 19 kno(i,knomax)= 21
i= 148 x= 4 y= 17 Hunger= 1 Fatigue= 0 Companionship= 65 Activity= 58 QOL= 97
kno(i,1)= 2 kno(i,2)= 16 kno(i,3)= 16 kno(i,knomax)= 21
i= 149 x= 3 y= 3 Hunger= 118 Fatigue= 45 Companionship= 1 Activity= 99 QOL= 181.9
kno(i,1)= 1 kno(i,2)= 3 kno(i,3)= 3 kno(i,knomax)= 21
i= 150 x= 17 y= 18 Hunger= 85 Fatigue= 25 Companionship= 15 Activity= 40 QOL= 319.7
kno(i,1)= 1 kno(i,2)= 19 kno(i,3)= 19 kno(i,knomax)= 21
i= 151 x= 4 y= 4 Hunger= 83 Fatigue= 35 Companionship= 0 Activity= 68 QOL= 370.8
kno(i,1)= 1 kno(i,2)= 16 kno(i,3)= 6 kno(i,knomax)= 21
i= 152 x= 18 y= 13 Hunger= 20 Fatigue= 13 Companionship= 29 Activity= 14 QOL= 447
kno(i,1)=-1 kno(i,2)= 16 kno(i,3)= 16 kno(i,knomax)= 21
i= 153 x= 3 y= 9 Hunger= 24 Fatigue= 33 Companionship= 9 Activity= 93 QOL= 300.3
kno(i,1)=-1 kno(i,2)= 6 kno(i,3)= 6 kno(i,knomax)= 21
i= 154 x= 19 y= 12 Hunger= 14 Fatigue= 40 Companionship= 35 Activity= 94 QOL= 462.6
kno(i,1)=-6 kno(i,2)= 19 kno(i,3)= 11 kno(i,knomax)= 21
i= 155 x= 16 y= 19 Hunger= 89 Fatigue= 7 Companionship= 31 Activity= 156 QOL=-80.5
kno(i,1)= 8 kno(i,2)= 16 kno(i,3)= 19 kno(i,knomax)= 21
i= 156 x= 13 y= 13 Hunger= 146 Fatigue= 58 Companionship= 17 Activity= 111 QOL=-58.5
kno(i,1)= 1 kno(i,2)= 16 kno(i,3)= 16 kno(i,knomax)= 21
i= 157 x= 6 y= 6 Hunger= 4 Fatigue= 0 Companionship= 8 Activity= 12 QOL= 633.8
kno(i,1)= 2 kno(i,2)= 11 kno(i,3)= 11 kno(i,knomax)= 21
i= 158 x= 12 y= 15 Hunger= 26 Fatigue= 31 Companionship= 25 Activity= 24 QOL= 599.3
kno(i,1)= 2 kno(i,2)= 11 kno(i,3)= 16 kno(i,knomax)= 21
i= 159 x= 17 y= 6 Hunger= 67 Fatigue= 68 Companionship= 36 Activity= 118 QOL=
667.500000000001
kno(i,1)=-6 kno(i,2)= 19 kno(i,3)= 3 kno(i,knomax)= 21
i= 160 x= 17 y= 18 Hunger= 124 Fatigue= 35 Companionship= 15 Activity= 100 QOL= 287.1
kno(i,1)= 1 kno(i,2)= 19 kno(i,3)= 19 kno(i,knomax)= 21
i= 161 x= 10 y= 7 Hunger= 52 Fatigue= 45 Companionship= 32 Activity= 7 QOL= 667.6
kno(i,1)= 1 kno(i,2)= 11 kno(i,3)= 11 kno(i,knomax)= 21
i= 162 x= 3 y= 4 Hunger= 41 Fatigue= 34 Companionship= 0 Activity= 7 QOL= 3036.1
kno(i,1)= 1 kno(i,2)= 3 kno(i,3)= 3 kno(i,knomax)= 21
i= 163 x= 16 y= 15 Hunger= 81 Fatigue= 82 Companionship= 7 Activity= 46 QOL= 902.9
kno(i,1)= 2 kno(i,2)= 16 kno(i,3)= 16 kno(i,knomax)= 21
i= 164 x= 10 y= 11 Hunger= 1 Fatigue= 11 Companionship= 26 Activity= 62 QOL= 843.3
kno(i,1)=-1 kno(i,2)= 6 kno(i,3)= 6 kno(i,knomax)= 21
i= 165 x= 14 y= 16 Hunger= 23 Fatigue= 24 Companionship= 15 Activity= 23 QOL= 589.5
kno(i,1)= 2 kno(i,2)= 16 kno(i,3)= 16 kno(i,knomax)= 21
i= 166 x= 17 y= 6 Hunger= 0 Fatigue= 5 Companionship= 119 Activity= 83 QOL= 364.2
kno(i,1)= 1 kno(i,2)= 17 kno(i,3)= 6 kno(i,knomax)= 21
i= 167 x= 12 y= 16 Hunger= 18 Fatigue= 19 Companionship= 14 Activity= 18 QOL= 3307.4
kno(i,1)= 2 kno(i,2)= 11 kno(i,3)= 16 kno(i,knomax)= 21
i= 168 x= 17 y= 6 Hunger= 33 Fatigue= 36 Companionship= 117 Activity= 86 QOL= 744.6
kno(i,1)= 7 kno(i,2)= 17 kno(i,3)= 6 kno(i,knomax)= 21
i= 169 x= 6 y= 2 Hunger= 49 Fatigue= 50 Companionship= 46 Activity= 49 QOL= 1331.9

```
    kno(i,1)= 2  kno(i,2)= 6  kno(i,3)= 6  kno(i,knomax)= 21
i= 170  x= 10  y= 6  Hunger= 100  Fatigue= 1  Companionship= 0  Activity= 100  QOL= 2300.3
    kno(i,1)= 7  kno(i,2)= 11  kno(i,3)= 6  kno(i,knomax)= 21
i= 171  x= 4  y= 5  Hunger= 105  Fatigue= 32  Companionship= 0  Activity= 95  QOL= 1946.4
    kno(i,1)= 1  kno(i,2)= 6  kno(i,3)= 16  kno(i,knomax)= 21
i= 172  x= 3  y= 5  Hunger= 120  Fatigue= 61  Companionship= 0  Activity= 92  QOL= 1780.3
    kno(i,1)= 1  kno(i,2)= 6  kno(i,3)= 16  kno(i,knomax)= 21
i= 173  x= 16  y= 6  Hunger= 34  Fatigue= 47  Companionship= 15  Activity= 44  QOL= 1336.5
    kno(i,1)= 2  kno(i,2)= 16  kno(i,3)= 11  kno(i,knomax)= 21
i= 174  x= 3  y= 3  Hunger= 108  Fatigue= 30  Companionship= 1  Activity= 90  QOL= 2522.9
    kno(i,1)= 1  kno(i,2)= 3  kno(i,3)= 3  kno(i,knomax)= 21
i= 175  x= 3  y= 6  Hunger= 74  Fatigue= 75  Companionship= 23  Activity= 51  QOL= 3076
    kno(i,1)= 2  kno(i,2)= 6  kno(i,3)= 6  kno(i,knomax)= 21
i= 176  x= 6  y= 2  Hunger= 26  Fatigue= 27  Companionship= 25  Activity= 26  QOL= 3930.4
    kno(i,1)= 2  kno(i,2)= 6  kno(i,3)= 6  kno(i,knomax)= 21
i= 177  x= 16  y= 6  Hunger= 31  Fatigue= 40  Companionship= 15  Activity= 39  QOL= 2222.8
    kno(i,1)= 2  kno(i,2)= 16  kno(i,3)= 11  kno(i,knomax)= 21
i= 178  x= 3  y= 5  Hunger= 126  Fatigue= 33  Companionship= 0  Activity= 82  QOL= 1723
    kno(i,1)= 1  kno(i,2)= 6  kno(i,3)= 16  kno(i,knomax)= 21
i= 179  x= 16  y= 4  Hunger= 3  Fatigue= 36  Companionship= 3  Activity= 33  QOL= 1567.9
    kno(i,1)= 2  kno(i,2)= 16  kno(i,3)= 11  kno(i,knomax)= 21
i= 180  x= 3  y= 5  Hunger= 126  Fatigue= 31  Companionship= 0  Activity= 97  QOL= 360.3
    kno(i,1)= 1  kno(i,2)= 6  kno(i,3)= 16  kno(i,knomax)= 21
i= 181  x= 16  y= 19  Hunger= 108  Fatigue= 78  Companionship= 10  Activity= 102  QOL= 3160.2
    kno(i,1)= 2  kno(i,2)= 13  kno(i,3)= 19  kno(i,knomax)= 21
i= 182  x= 2  y= 10  Hunger= 99  Fatigue= 41  Companionship= 0  Activity= 98  QOL= 2639.5
    kno(i,1)= 8  kno(i,2)= 2  kno(i,3)= 10  kno(i,knomax)= 21
i= 183  x= 13  y= 12  Hunger= 182  Fatigue= 0  Companionship= 17  Activity= 126  QOL= 1808.2
    kno(i,1)= 2  kno(i,2)= 16  kno(i,3)= 11  kno(i,knomax)= 21
i= 184  x= 12  y= 15  Hunger= 30  Fatigue= 32  Companionship= 26  Activity= 30  QOL= 1788.9
    kno(i,1)= 8  kno(i,2)= 12  kno(i,3)= 15  kno(i,knomax)= 21
i= 185  x= 6  y= 6  Hunger= 73  Fatigue= 0  Companionship= 20  Activity= 72  QOL= 1751.6
    kno(i,1)= 2  kno(i,2)= 6  kno(i,3)= 5  kno(i,knomax)= 21
i= 186  x= 5  y= 6  Hunger= 12  Fatigue= 13  Companionship= 4  Activity= 12  QOL= 1968.2
    kno(i,1)= 2  kno(i,2)= 6  kno(i,3)= 6  kno(i,knomax)= 21
i= 187  x= 2  y= 6  Hunger= 99  Fatigue= 0  Companionship= 73  Activity= 99  QOL= 650.1
    kno(i,1)= 7  kno(i,2)= 6  kno(i,3)= 6  kno(i,knomax)= 21
i= 188  x= 16  y= 18  Hunger= 82  Fatigue= 36  Companionship= 15  Activity= 98  QOL= 2371.2
    kno(i,1)=-1  kno(i,2)= 6  kno(i,3)= 6  kno(i,knomax)= 21
i= 189  x= 11  y= 11  Hunger= 84  Fatigue= 11  Companionship= 10  Activity= 77  QOL= 1004.8
    kno(i,1)= 1  kno(i,2)= 11  kno(i,3)= 11  kno(i,knomax)= 21
i= 190  x= 16  y= 3  Hunger= 44  Fatigue= 6  Companionship= 0  Activity= 104  QOL= 1647.2
    kno(i,1)= 8  kno(i,2)= 16  kno(i,3)= 3  kno(i,knomax)= 21
i= 191  x= 3  y= 6  Hunger= 90  Fatigue= 77  Companionship= 69  Activity= 21  QOL= 3717.2
    kno(i,1)= 2  kno(i,2)= 6  kno(i,3)= 6  kno(i,knomax)= 21
i= 192  x= 6  y= 11  Hunger= 12  Fatigue= 0  Companionship= 3  Activity= 11  QOL= 1556.9
    kno(i,1)= 2  kno(i,2)= 6  kno(i,3)= 11  kno(i,knomax)= 21
i= 193  x= 10  y= 6  Hunger= 45  Fatigue= 46  Companionship= 0  Activity= 45  QOL= 3616.5
    kno(i,1)= 2  kno(i,2)= 11  kno(i,3)= 6  kno(i,knomax)= 21
i= 194  x= 12  y= 16  Hunger= 35  Fatigue= 50  Companionship= 18  Activity= 48  QOL= 1105.9
    kno(i,1)= 2  kno(i,2)= 11  kno(i,3)= 16  kno(i,knomax)= 21
i= 195  x= 4  y= 4  Hunger= 92  Fatigue= 0  Companionship= 46  Activity= 98  QOL= 1000.9
    kno(i,1)= 2  kno(i,2)= 6  kno(i,3)= 6  kno(i,knomax)= 21
i= 196  x= 6  y= 6  Hunger= 28  Fatigue= 20  Companionship= 2  Activity= 21  QOL= 1918
    kno(i,1)= 1  kno(i,2)= 11  kno(i,3)= 11  kno(i,knomax)= 21
i= 197  x= 19  y= 13  Hunger= 22  Fatigue= 1  Companionship= 7  Activity= 7  QOL= 1625.2
```

kno(i,1)= 1 kno(i,2)= 19 kno(i,3)= 19 kno(i,knomax)= 21
i= 198 x= 12 y= 11 Hunger= 93 Fatigue= 68 Companionship= 10 Activity= 127 QOL= 384.3
kno(i,1)=-1 kno(i,2)= 16 kno(i,3)= 16 kno(i,knomax)= 21
i= 199 x= 2 y= 21 Hunger= 41 Fatigue= 65 Companionship= 19 Activity= 70 QOL= 2126.2
kno(i,1)= 8 kno(i,2)= 2 kno(i,3)= 21 kno(i,knomax)= 21
i= 200 x= 18 y= 13 Hunger= 20 Fatigue= 13 Companionship= 29 Activity= 8 QOL= 1535.6
kno(i,1)=-1 kno(i,2)= 16 kno(i,3)= 16 kno(i,knomax)= 21

Population at end of run = 110

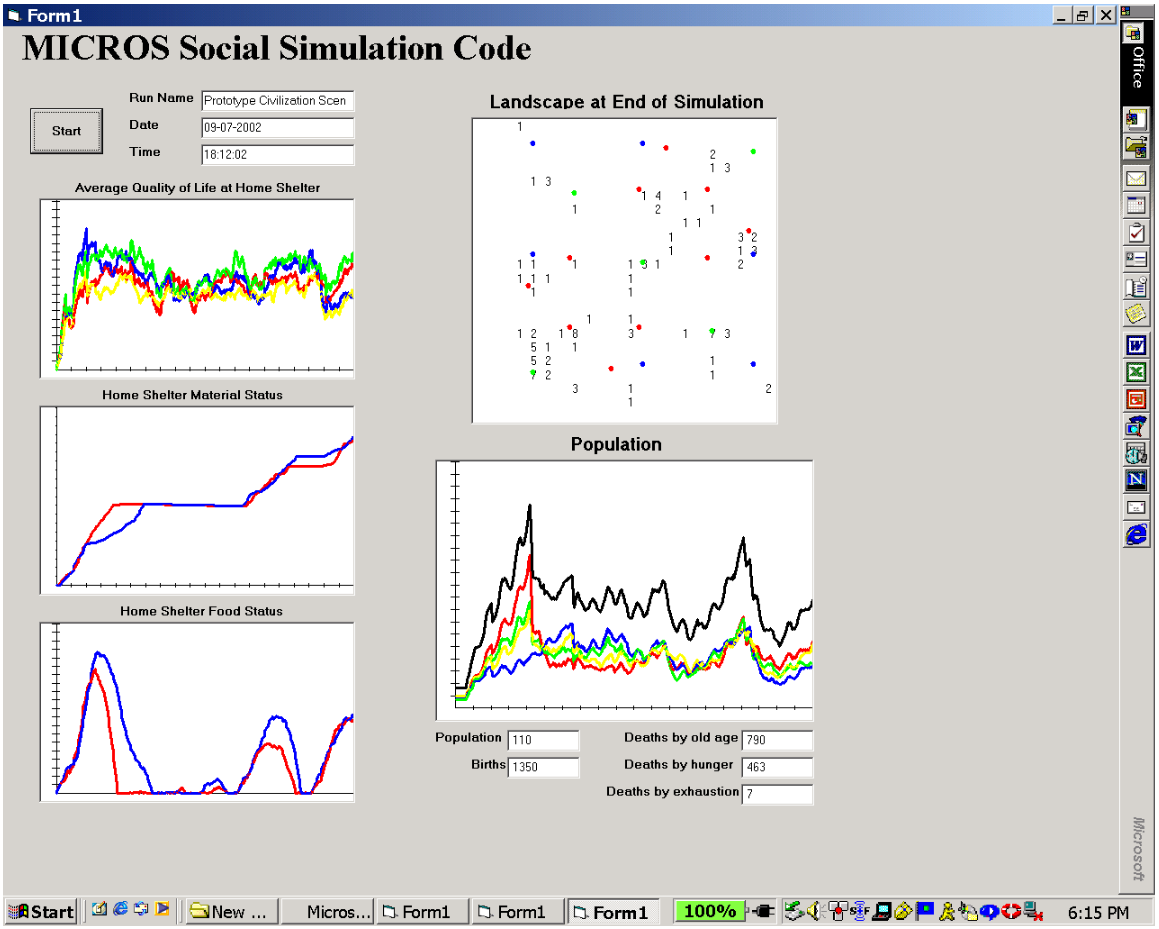
Births = 1350
Deaths = 1260
Deaths by old age = 790
Deaths by hunger = 463
Deaths by exhaustion = 7

Time average of number of good agents= 50.572925
Time average of quality of life of good agents= 1708.18952957806
Time average of strength of good agents= 874.477109156522
Time average of power factor of good agents= 34947.1051856029
Time average of social factor of good agents= 33572.3579402707
Time average of number of communications instituted by good agents= 4.41405678816571
Time average of number of sharings instituted by good agents= 8.02034003184976E-03

Time average of number of bad agents= 49.729725
Time average of quality of life of bad agents= 1309.23298810075
Time average of strength of bad agents= 882.194528534573
Time average of power factor of bad agents= 2338.18358940991
Time average of social factor of bad agents= 1933.78645984371
Time average of number of communications instituted by bad agents= 0
Time average of number of stealing events instituted by bad agents= 3.7736211593986E-03

Time average of number of matings involving two good agents= 3.37227480791214E-06
Time average of number of matings involving one good and one bad agent= 7.51497731192261E-06
Time average of number of matings involving two bad agents= 2.76986406657383E-06

Figure 1. Screen Plots For Run With Input File Given in Table 4.



VI. Future Options for MICROS

There are many interesting studies that can be performed with this version of MICROS. These include:

1. A Paradise Scenario with all agents set to good, no leaders, no reproduction (gestation set to a very large value), a long lifetime, and well-supplied food centers.
2. A Jungle Scenario with all agents set to bad, no leaders, reproduction without monogamy and severely limited food centers.
3. A Proto-civilization Scenario with a mix of good and bad agents, leaders, reproduction with monogamy and modestly stocked food centers.

Minor changes in the code would allow more complex social structures to be simulated. An Agrarian Scenario could be constructed in which the productivity of nearby food centers would be determined by the material status of the home shelter. Successful home shelters could support a greater population. A Conflict Scenario, in which one home shelter competed with and engaged in warfare with the other home shelter, would be a natural extension of the introduction of violence into agent personality.

Additional opportunities for good and evil will help refine and expand the computational approach to ethics that is the motivation for this work. In particular, lying (intentional miscommunication of information), revenge based on personal, family, or shelter based affronts might be considered. Of particular interest is the effect of historical (multi-generational) ill-will of one shelter population toward another or the bias of one population sub-group toward another (racism).

Some of these and other simulations will be described in the next report in this series.

Reference

S.M. Younger (2002) *Discrete Agent Modeling as a Tool for the Study of Individual and Social Development: Initial Studies*. Los Alamos National Laboratory Report LA-UR-02-6698.

Government Report Disclaimer